### Deadlines

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Date of assignment</th>
<th>Deadline (midnight 23:59)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW1</td>
<td>Sep 6</td>
<td>Oct 19</td>
</tr>
<tr>
<td>HW2</td>
<td>Sep 20</td>
<td>Oct 3</td>
</tr>
<tr>
<td>HW3</td>
<td>Oct 4</td>
<td>Oct 17</td>
</tr>
<tr>
<td>Paper summary</td>
<td>Oct 11</td>
<td>Oct 31</td>
</tr>
<tr>
<td>HW4</td>
<td>Oct 18</td>
<td>Oct 31</td>
</tr>
<tr>
<td>HW5</td>
<td>Nov 1</td>
<td>Nov 14</td>
</tr>
<tr>
<td>HW6</td>
<td>Nov 22</td>
<td>Dec 5</td>
</tr>
<tr>
<td>Project</td>
<td>Projects are collected</td>
<td>Dec 13 - 15</td>
</tr>
</tbody>
</table>

*All deadlines are subject to change, check out CampusWire and website for updates*
What is deep learning?

An adaptive **non-linear mapping** from one **space** to another

\[ \text{Space of things} \rightarrow \text{Artificial Neural Networks with many layers} \rightarrow \text{Space of labels} \]

\[ \text{DL} = \text{Networks with many layers} \]
Let's calculate **class scores** for each class using **feed-forward path** algorithm.

\[
\begin{align*}
\text{out}(h_1) &= 0.59 \\
\text{in}(o_1) &= 1.1 \\
\text{out}(o_1) &= 0.75 \\
\text{out}(o_2) &= 0.77
\end{align*}
\]
Backpropagation algorithm

Let's try to reduce total error by changing \( w_5 \)

We need to find a gradient \( \frac{\partial E_{\text{total}}}{\partial w_5} \) with respect to \( w_5 \)

\[
\frac{\partial E_{\text{total}}}{\partial w_5} = \frac{\partial E_{\text{total}}}{\partial E_{o_1}} \times \frac{\partial E_{o_1}}{\partial o_{out_1}} \times \frac{\partial o_{out_1}}{\partial o_{in_1}} \times \frac{\partial o_{in_1}}{\partial w_5}
\]
Fei-Fei Li

1.2 million images

1000 categories
Errors

<table>
<thead>
<tr>
<th>Year</th>
<th>2010</th>
<th>2011</th>
<th>2012</th>
</tr>
</thead>
<tbody>
<tr>
<td>Errors</td>
<td>28%</td>
<td>26%</td>
<td>16%</td>
</tr>
</tbody>
</table>

AlexNet (A. Krizhevsky et al. 2012)

http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/
Errors

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Rate</td>
<td>28%</td>
<td>26%</td>
<td>16%</td>
<td>12%</td>
<td>7%</td>
<td>3%</td>
<td>&lt;3%</td>
</tr>
</tbody>
</table>

AlexNet (A. Krizhevsky et al. 2012)

http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/
Hypothetical super-dedicated fine-grained expert ensemble of human labelers

AlexNet (A. Krizhevsky et al. 2012)

http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/
The following slides are adapted from the YouTube lecture by Brandon Rohrer at https://youtu.be/FmpDIaiM1eA.
Categorise images

81 pixel image
Categorise images

81 pixel image

X
O
Categorise images

81 pixel image

X

O
Categorise images

![Diagram showing the process of categorising images using CNN. The images are processed by CNN, resulting in a classification of 'X' and 'O'.]
It needs to handle **natural variety** in the input data.
Are these the same images?
For the algorithm this is **not** straightforward
For the algorithm this is **not** straight forward.
For the algorithm this is **not** straight forward
For the algorithm this is not straight forward

Computer can end up concluding that these two images contain different patterns
Convolutional Neural Networks
compare smaller patterns
Convolutional Neural Networks
compare smaller patterns
Convolutional Neural Networks
compare smaller patterns
Convolutional Neural Networks
compare smaller patterns
Convolutional Neural Networks compare smaller patterns.

This helps to quantify the similarity between slightly different images.
Convolutional Neural Networks
compare smaller patterns
Convolutional Neural Networks

compare smaller patterns
Convolutional Neural Networks
compare smaller patterns
Filtering: locating patterns on the image
Filtering: locating patterns on the image

Resulting matrix

\[ \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \]
Filtering: locating patterns on the image

Resulting matrix

\[ \begin{pmatrix} 1 \\ \vdots \end{pmatrix} \]
Filtering: locating patterns on the image

Resulting matrix

\[ -1 \times -1 = 1 \]
Filtering: locating patterns on the image

\[-1 \times -1 = 1\]

Resulting matrix

\[
\begin{bmatrix}
1 & 1 & 1 \\
\end{bmatrix}
\]
Filtering: locating patterns on the image

Resulting matrix

\[
\begin{bmatrix}
 1 & 1 & 1 \\
 1 & 1 & 1 \\
 1 & 1 & 1 \\
\end{bmatrix}
\]
Filtering: locating patterns on the image

Resulting matrix

\[
\begin{pmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{pmatrix}
\]
Filtering: locating patterns on the image

Resulting matrix

\[
\begin{array}{ccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{array}
\]

\[
\frac{1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1}{9} = 1
\]
Filtering: locating patterns on the image

Resulting matrix

\[
\begin{array}{ccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{array}
\]

\[
\frac{1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1}{9} = 1
\]
Filtering: locating patterns on the image
Filtering: locating patterns on the image

Resulting matrix

\[
\begin{bmatrix}
1 \\
\end{bmatrix}
\]
Filtering: locating patterns on the image

\[ \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{pmatrix} \times \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} = 1 

Resulting matrix:

\[ \begin{pmatrix} 1 & 1 \\ \vdots & \vdots \\ \vdots & \vdots \end{pmatrix} \]
Filtering: locating patterns on the image

\[
\begin{pmatrix}
1 & -1 & -1 \\
-1 & 1 & -1 \\
-1 & -1 & 1
\end{pmatrix}
\times
\begin{pmatrix}
1
\end{pmatrix} = -1
\]
Filtering: locating patterns on the image

Resulting matrix

\[
\begin{pmatrix}
1 & 1 & -1 \\
1 & 1 & 1 \\
-1 & 1 & 1 \\
\end{pmatrix}
\]
Filtering: locating patterns on the image

Resulting matrix

\[
\begin{array}{ccc}
1 & 1 & -1 \\
1 & 1 & 1 \\
-1 & 1 & 1 \\
\end{array}
\]

\[
\frac{1 + 1 - 1 + 1 + 1 + 1 - 1 + 1 + 1}{9} = 0.55
\]
Filtering: trying all possible matches
What is going on on the edges?
What is going on on the edges?

1) we can fill in missing values with 0s (zero padding)
What is going on on the **edges**?

1) we can fill in missing values with **0s** (zero padding)
2) we can mirror pixels from the original image (mirror padding)
What is going on on the edges?

1) we can fill in missing values with 0s (zero padding)
2) we can mirror pixels from the original image (mirror padding)
3) ignore them
Convolutional operation

\[-1 \quad -1 \quad -1 \quad -1\]
\[-1 \quad 1 \quad -1 \quad -1\]
\[-1 \quad -1 \quad 1 \quad -1\]
\[-1 \quad -1 \quad -1 \quad 1\]
\[-1 \quad 1 \quad -1 \quad -1\]
\[-1 \quad -1 \quad 1 \quad -1\]
\[-1 \quad -1 \quad -1 \quad 1\]
\[-1 \quad 1 \quad -1 \quad -1\]
\[-1 \quad -1 \quad 1 \quad -1\]

\[1 \quad -1 \quad -1\]
\[-1 \quad 1 \quad -1\]
\[-1 \quad -1 \quad 1\]

\[0.77 \quad -0.11 \quad 0.11 \quad 0.33 \quad 0.55 \quad -0.11 \quad 0.33\]
\[-0.11 \quad 1 \quad -0.11 \quad 0.33 \quad -0.11 \quad 0.11 \quad -0.11\]
\[0.11 \quad -0.11 \quad 1 \quad -0.33 \quad 0.11 \quad -0.11 \quad 0.55\]
\[0.33 \quad 0.33 \quad -0.33 \quad 0.55 \quad -0.33 \quad 0.33 \quad 0.33\]
\[0.55 \quad -0.11 \quad 0.11 \quad -0.33 \quad 1 \quad -0.11 \quad 0.11\]
\[-0.11 \quad 0.11 \quad -0.11 \quad 0.33 \quad -0.11 \quad 1 \quad -0.11\]
\[0.33 \quad -0.11 \quad 0.55 \quad 0.33 \quad 0.11 \quad -0.11 \quad 0.77\]
These filters are trainable through backpropagation.
Depending on the problem convolutional filters may learn different patterns.
Convolutional operation
Convolutional operation

Convolutional layer
Applying **activation function** to results of convolutional layer
Applying **activation function** to results of convolutional layer
Applying **activation function** to results of convolutional layer
Applying **activation function** to results of convolutional layer

**ReLu activation**

\[ max(0, x) \]
Applying **activation function** to results of convolutional layer

\[
\begin{array}{cccccccc}
0.33 & -0.55 & 0.11 & -0.11 & 0.11 & -0.55 & 0.33 \\
-0.55 & 0.55 & -0.55 & 0.33 & -0.55 & 0.55 & -0.55 \\
0.11 & -0.55 & 0.55 & -0.77 & 0.55 & -0.55 & 0.55 \\
-0.11 & 0.33 & -0.77 & 1 & -0.77 & 0.33 & -0.11 \\
0.11 & -0.55 & 0.55 & -0.77 & 0.55 & -0.55 & 0.11 \\
-0.55 & 0.55 & -0.55 & 0.33 & -0.55 & 0.55 & -0.55 \\
0.33 & -0.55 & 0.55 & -0.11 & 0.11 & -0.55 & 0.33 \\
\end{array}
\]

ReLu activation

\[\text{max}(0, x)\]
Applying **activation function** to results of convolutional layer

![Activation Function Diagram](image-url)
Applying **activation function** to results of convolutional layer.

ReLu activation:

$$\text{max}(0, x)$$
Applying **activation function** to results of convolutional layer
Very often convolutional layer is followed by ReLu activations.
Very often convolutional layer is followed by ReLu activations

This whole thing should remind you of a regular neuron that we saw before
Classical artificial neuron
from previous lecture
**Classical artificial neuron** from previous lecture

**Convolutional filter with Sigmoid**
Classical artificial neuron from previous lecture

Convolutional filter with Sigmoid
Classical artificial neuron from previous lecture

Convolutional filter with Sigmoid
Classical artificial neuron from previous lecture

Convolutional filter with Sigmoid
**Classical** artificial neuron from previous lecture

**Convolutional** filter with Sigmoid
Classical artificial neuron from previous lecture

Convolutional filter with Sigmoid
Classical artificial neuron from previous lecture

Convolutional filter with Sigmoid

Results in 4 values
Classical artificial neuron from previous lecture

Convolutional filter with Sigmoid

Results in 4 values
Classical artificial neuron from previous lecture

Convolutional filter with Sigmoid
**Classical** artificial neuron from previous lecture

What about the **rest** of the **weights**?

**Convolutional** filter with **Sigmoid**
Classical artificial neuron from previous lecture

Convolutional filter with Sigmoid
Classical artificial neuron from previous lecture

Convolutional filter with Sigmoid
Classical artificial neuron from previous lecture

Convolutional filter with Sigmoid
Classical artificial neuron from previous lecture

Convolutional filter with Sigmoid

One would need one neuron per each filtering operation to replicate convolutional layer
Very often convolutional layer is followed by ReLu activations
Reducing the dimensionality by **pooling**
Reducing the dimensionality by pooling
Reducing the dimensionality by pooling

Take maximum out of 2x2
Reducing the dimensionality by pooling

Take maximum out of 2x2
Reducing the dimensionality by **pooling**

Take maximum out of 2x2
Reducing the dimensionality by **pooling**

Pooling helps to reduce dimensionality preserving pattern (not always safe)
Stacking all of these together

Convolutional layer + ReLu layer → Pooling layer
Stacking all of these together
Making even more combinations of layers
Making even more combinations of layers
Making even more combinations of layers
Making even more combinations of layers
Making even more combinations of layers

Convolutional filters depending on the layer **capture** more and more **sophisticated** patterns
Making even more combinations of layers

Convolutional filters depending on the layer capture more and more sophisticated patterns
From Visualizing and Understanding Convolutional Networks (https://arxiv.org/pdf/1311.2901.pdf)
Final layer is **fully connected** layer.
Finally the full **convolutional network**:
Finally the full convolutional network:

Something similar to AlexNet:
Training these networks the same way as we saw before using **backpropagation**.
Camera(s)

- Measurements (e.g. current speed and steering)
- LiDAR (optional)
- HD Maps (optional)

Route planner

Behavior

Measurements

- CNN
- MLP

Pre-Net (e.g. PointPillars)

LiDAR

CNN (e.g. ResNet)

HD Maps

Segmentation

- Human
- Car
- Depth

Object detection

- Car
- Optical flow

Costmap

Auxiliary outputs

- Waypoints
- Actuation (e.g. steering, acc., break)

Switching heads of MLP

RNN (e.g. LSTM)

MLP

A mostly complete chart of

Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org

**Forward pass:**

<table>
<thead>
<tr>
<th>$X_{11}$</th>
<th>$X_{12}$</th>
<th>$X_{13}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_{21}$</td>
<td>$X_{22}$</td>
<td>$X_{23}$</td>
</tr>
<tr>
<td>$X_{31}$</td>
<td>$X_{32}$</td>
<td>$X_{33}$</td>
</tr>
</tbody>
</table>

```
W_{11}   W_{12}

W_{21}   W_{22}
```

```
```
Forward pass:
Forward pass:

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33} \\
\end{array}
\]
Forward pass:

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33}
\end{array}
\]

\[
\begin{array}{cc}
W_{11} & W_{12} \\
W_{21} & W_{22}
\end{array}
\]

\[
\begin{array}{c}
h_{11} \\
W_{11}X_{12} + W_{12}X_{13} + W_{21}X_{22} + W_{22}X_{23}
\end{array}
\]
Forward pass:

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33} \\
\end{array}
\]

\[
\begin{array}{cc}
W_{11} & W_{12} \\
W_{21} & W_{22} \\
\end{array}
\]

\[
\begin{array}{cc}
h_{11} & h_{12} \\
\end{array}
\]

\[
\begin{array}{c}
W_{11}X_{21} + W_{12}X_{22} + W_{21}X_{31} + W_{22}X_{32} \\
\end{array}
\]
Forward pass:

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33}
\end{array}
\]

\[
\begin{array}{cc}
W_{11} & W_{12} \\
W_{21} & W_{22}
\end{array}
\]

\[
\begin{array}{cc}
h_{11} & h_{12} \\
h_{21}
\end{array}
\]
Forward pass:

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33}
\end{array}
\]

\[
\begin{array}{cc}
W_{11} & W_{12} \\
W_{21} & W_{22}
\end{array}
\]

\[
\begin{array}{cc}
h_{11} & h_{12} \\
h_{21} & W_{11}X_{22} + W_{12}X_{23} + W_{21}X_{32} + W_{22}X_{33}
\end{array}
\]
Forward pass:

\[
\begin{bmatrix}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33}
\end{bmatrix}
\]

\[
\begin{bmatrix}
W_{11} & W_{12} \\
W_{21} & W_{22}
\end{bmatrix}
\]

\[
\begin{bmatrix}
h_{11} & h_{12} \\
h_{21} & h_{22}
\end{bmatrix}
\]
Forward pass:

<table>
<thead>
<tr>
<th>X_{11}</th>
<th>X_{12}</th>
<th>X_{13}</th>
</tr>
</thead>
<tbody>
<tr>
<td>X_{21}</td>
<td>X_{22}</td>
<td>X_{23}</td>
</tr>
<tr>
<td>X_{31}</td>
<td>X_{32}</td>
<td>X_{33}</td>
</tr>
</tbody>
</table>

\[
\begin{array}{cc}
W_{11} & W_{12} \\
W_{21} & W_{22}
\end{array}
\]

<table>
<thead>
<tr>
<th>h_{11}</th>
<th>h_{12}</th>
</tr>
</thead>
<tbody>
<tr>
<td>h_{21}</td>
<td>h_{22}</td>
</tr>
</tbody>
</table>
Forward pass:

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33}
\end{array}
\]

\[
\begin{array}{cc}
W_{11} & W_{12} \\
W_{21} & W_{22}
\end{array}
\]

\[
\begin{array}{cc}
h_{11} & h_{12} \\
h_{21} & h_{22}
\end{array}
\]

Backward pass:

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33}
\end{array}
\]

\[
\begin{array}{cc}
W_{11} & W_{12} \\
W_{21} & W_{22}
\end{array}
\]

\[
\begin{array}{cc}
h_{11} & h_{12} \\
h_{21} & h_{22}
\end{array}
\]


**Forward pass:**

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33}
\end{array}
\]

**Backward pass:**

\[
\begin{array}{c}
W_{11} \\
W_{21} \\
W_{12} \\
W_{22}
\end{array}
\]
**Forward pass:**

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33}
\end{array}
\]

\[
\begin{array}{cc}
W_{11} & W_{12} \\
W_{21} & W_{22}
\end{array}
\]

\[
\begin{array}{cc}
h_{11} & h_{12} \\
h_{21} & h_{22}
\end{array}
\]

**Backward pass:**

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33}
\end{array}
\]

\[
\begin{array}{cc}
dh_{11} & dh_{12} \\
dh_{21} & dh_{22}
\end{array}
\]
**Forward pass:**

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33}
\end{array}
\]

\[
\begin{array}{cc}
W_{11} & W_{12} \\
W_{21} & W_{22}
\end{array}
\]

\[
\begin{array}{ccc}
h_{11} & h_{12} \\
h_{21} & h_{22}
\end{array}
\]

**Backward pass:**

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33}
\end{array}
\]

\[
\begin{array}{cc}
\text{dh}_{11}X_{11} & \text{dh}_{12}X_{12} \\
\text{dh}_{21}X_{21} & \text{dh}_{22}X_{22}
\end{array}
\]

\[
\begin{array}{ccc}
\text{dh}_{11} & \text{dh}_{12} \\
\text{dh}_{21} & \text{dh}_{22}
\end{array}
\]
Forward pass:

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33}
\end{array}
\]

Backward pass:

\[
\begin{array}{ccc}
\text{dh}_{11} & \text{dh}_{12} \\
\text{dh}_{21} & \text{dh}_{22}
\end{array}
\]
Forward pass:

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33}
\end{array}
\]

\[
\begin{array}{cc}
W_{11} & W_{12} \\
W_{21} & W_{22}
\end{array}
\]

\[
\begin{array}{cc}
h_{11} & h_{12} \\
h_{21} & h_{22}
\end{array}
\]

Backward pass:

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33}
\end{array}
\]

\[
\begin{array}{cc}
dh_{11} & dh_{12} \\
dh_{21} & dh_{22}
\end{array}
\]

\[
\begin{array}{cc}
dh_{11}X_{11} + dh_{11}X_{21} & dh_{12}X_{12} + dh_{12}X_{13} + dh_{12}X_{22} \\
dh_{21}X_{21} + dh_{21}X_{31} & dh_{22}X_{22} + dh_{22}X_{23} + dh_{22}X_{32}
\end{array}
\]
**Forward** pass:

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33} \\
\end{array}
\]

\[
\begin{array}{cc}
W_{11} & W_{12} \\
W_{21} & W_{22} \\
\end{array}
\]

\[
\begin{array}{cc}
h_{11} & h_{12} \\
h_{21} & h_{22} \\
\end{array}
\]

**Backward** pass:

\[
\begin{array}{ccc}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23} \\
X_{31} & X_{32} & X_{33} \\
\end{array}
\]

\[
\begin{array}{cccc}
\text{dh}_{11}X_{11} + \\
\text{dh}_{11}X_{12} + \\
\text{dh}_{11}X_{21} + \\
\text{dh}_{11}X_{22} \\
\text{dh}_{21}X_{21} + \\
\text{dh}_{21}X_{22} + \\
\text{dh}_{21}X_{31} + \\
\text{dh}_{21}X_{32} \\
\text{dh}_{12}X_{12} + \\
\text{dh}_{12}X_{13} + \\
\text{dh}_{12}X_{22} + \\
\text{dh}_{12}X_{23} \\
\text{dh}_{22}X_{22} + \\
\text{dh}_{22}X_{23} + \\
\text{dh}_{22}X_{32} + \\
\text{dh}_{22}X_{33} \\
\end{array}
\]

\[
\begin{array}{cc}
dh_{11} & dh_{12} \\
dh_{21} & dh_{22} \\
\end{array}
\]
### Forward pass:

<table>
<thead>
<tr>
<th></th>
<th>$X_{11}$</th>
<th>$X_{12}$</th>
<th>$X_{13}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_{21}$</td>
<td>$X_{22}$</td>
<td>$X_{23}$</td>
<td></td>
</tr>
<tr>
<td>$X_{31}$</td>
<td>$X_{32}$</td>
<td>$X_{33}$</td>
<td></td>
</tr>
</tbody>
</table>

### Backward pass:

<table>
<thead>
<tr>
<th></th>
<th>$X_{11}$</th>
<th>$X_{12}$</th>
<th>$X_{13}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_{21}$</td>
<td>$X_{22}$</td>
<td>$X_{23}$</td>
<td></td>
</tr>
<tr>
<td>$X_{31}$</td>
<td>$X_{32}$</td>
<td>$X_{33}$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>dh$_{11}$</th>
<th>dh$_{12}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dh$<em>{11}X</em>{11}$ + dh$<em>{11}X</em>{12}$ + dh$<em>{11}X</em>{21}$ + dh$<em>{11}X</em>{22}$</td>
<td>dh$<em>{12}X</em>{12}$ + dh$<em>{12}X</em>{13}$ + dh$<em>{12}X</em>{22}$ + dh$<em>{12}X</em>{23}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>dh$_{21}$</th>
<th>dh$_{22}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dh$<em>{21}X</em>{21}$ + dh$<em>{21}X</em>{22}$ + dh$<em>{21}X</em>{31}$ + dh$<em>{21}X</em>{32}$</td>
<td>dh$<em>{22}X</em>{22}$ + dh$<em>{22}X</em>{23}$ + dh$<em>{22}X</em>{32}$ + dh$<em>{22}X</em>{33}$</td>
</tr>
</tbody>
</table>
Dataset

Dogs
Vs
Cats
Dataset
Dataset
Dataset
Dataset

Validation

CNN

Learning Rate

Architecture

Optimizer
CNN Dataset

Batch #1

Dataset

Batch (size = 4)

Learning Rate

Architecture

Optimizer

CNN

prediction

Cat

Dog
CNN Dataset

Validation Dataset

Batch #1

Prediction

Cat

Dog

True

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
</tr>
</tbody>
</table>

CNN

Architecture

Optimizer

Learning Rate

Batch (size = 4)
Dataset

Batch (size = 4)

CNN

<table>
<thead>
<tr>
<th>Learning Rate</th>
<th>Architecture</th>
<th>Optimizer</th>
</tr>
</thead>
</table>

Table:

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
</tr>
</tbody>
</table>
CNN
Dataset
Batch (size = 4)

Validation Dataset

Batch #1

<table>
<thead>
<tr>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>0.84</td>
</tr>
<tr>
<td>Dog</td>
<td>0.65</td>
</tr>
<tr>
<td>0.38</td>
<td>0.46</td>
</tr>
</tbody>
</table>
Batch #1

Dataset

Learning Rate

Architecture

Optimizer

CNN

<table>
<thead>
<tr>
<th></th>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch</td>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>#1</td>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>

Binary cross-entropy error

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Binary cross-entropy error

\[ = - (y \log(p) + (1 - y) \log(1 - p)) \]

Dataset

Batch (size = 4)

Batch #1

Validation Dataset

Learning Rate, Architecture, Optimizer

CNN

prediction

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>Dog</td>
<td>True</td>
<td>Error</td>
</tr>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td></td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Dataset

CNN

Validation Dataset

Batch (size = 4)

Learning Rate

Architecture

Optimizer

Batch #1

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>Dog</td>
<td>True</td>
<td>Error</td>
</tr>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td></td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>

**Binary cross-entropy error**

\[
1 \text{ for cat} \quad \& \quad 0 \text{ for dog}
\]

\[
= - (y \log(p) + (1 - y) \log(1 - p))
\]

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Batch (size = 4) → CNN → prediction

### Binary cross-entropy error

\[ - (y \log(p) + (1 - y) \log(1 - p)) \]

1 for **cat** & 0 for **dog**

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td></td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>

Probability of class **cat**

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Binary cross-entropy error

\[ - (y \log(p) + (1 - y) \log(1 - p)) \]

1 for cat & 0 for dog

Probability of class cat

Probability of class dog

Dataset

CNN

Validation Dataset

Batch #1

batch #1

Learning Rate

Architecture

Optimizer

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>Dog</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td></td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>
Batch #1

Batch (size = 4)

Dataset

Learning Rate
Architecture
Optimizer

CNN

Validation Dataset

Batch #1

prediction

1 for cat & 0 for dog

Binary cross-entropy error

\[ \text{Probability of class cat} = - (y \log(p) + (1 - y) \log(1 - p)) \]

\[ \text{Probability of class dog} = \frac{1 - \text{Probability of class cat}}{2} \]

y = ?
p = ?

<table>
<thead>
<tr>
<th></th>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Binary cross-entropy error

$$1 \text{ for cat} & \quad 0 \text{ for dog}$$

$$y = 1$$

$$p = 0.84$$

$$= - (y \log(p) + (1 - y) \log(1 - p))$$

$$\text{Probability of class dog}$$

$$\text{Probability of class cat}$$

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td></td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Batch #1

Dataset

Batch (size = 4)

CNN

Validation Dataset

Learning Rate

Architecture

Optimizer

\[ p = 0.84 \quad y = 1 \]

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td></td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>

**Binary cross-entropy error**

\[ = -(1 \times \log(0.84) + (1 - 1) \times \log(1 - 0.84)) \]

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Binary cross-entropy error $= 0.25$

$p = 0.84$  \quad  y = 1$

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td></td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
# Batch #1

**Dataset**

**Batch** (size = 4)

**CNN**

**Learning Rate**

**Architecture**

**Optimizer**

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td>0.25</td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td></td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>

**Prediction**

\[ p = 0.84 \quad y = 1 \]

**Binary cross-entropy error**

\[ = 0.25 \]

[https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a](https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a)
Binary cross-entropy error

\[ \text{error} = - (1 \times \log(0.65) + (1 - 1) \times \log(1 - 0.65)) \]

<table>
<thead>
<tr>
<th></th>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction: 0.84</td>
<td>0.16</td>
<td>Cat</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>Prediction: 0.65</td>
<td>0.35</td>
<td>Cat</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prediction: 0.54</td>
<td>0.46</td>
<td>Dog</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prediction: 0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Dataset

Batch (size = 4)

CNN

Batch #1

Learning Rate

Architecture

Optimizer

Prediction

Binary cross-entropy error $= 0.62$

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td>0.25</td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td></td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td></td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>

$p = 0.65$ $y = 1$
Binary cross-entropy error = 0.62
Binary cross-entropy error

\[ \text{error} = - (0 \times \log(0.54) + (1 - 0) \times \log(1 - 0.54)) \]

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Binary cross-entropy error

\[ = - \log(1 - 0.54) \]

<table>
<thead>
<tr>
<th></th>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0.84</strong></td>
<td></td>
<td>0.16</td>
<td>Cat</td>
<td>0.25</td>
</tr>
<tr>
<td><strong>0.65</strong></td>
<td></td>
<td>0.35</td>
<td>Cat</td>
<td>0.62</td>
</tr>
<tr>
<td><strong>0.54</strong></td>
<td></td>
<td><strong>0.46</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>0.38</strong></td>
<td></td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>

\[ p = 0.54 \quad y = 0 \]
Binary cross-entropy error

\[ = - \log(0.46) \]

Dataset

Batch (size = 4)

CNN

Batch #1

Learning Rate

Architecture

Optimizer

p = 0.54, y = 0

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td>0.25</td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td>0.62</td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td></td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Batch (size = 4) → CNN

Learning Rate, Architecture, Optimizer

Dataset

Batch #1

Prediction

Binary cross-entropy error = 1.12

<table>
<thead>
<tr>
<th></th>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>y = 0</td>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td>0.25</td>
</tr>
<tr>
<td>y = 0</td>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td>0.62</td>
</tr>
<tr>
<td></td>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>

\[ p = 0.54 \]

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Dataset

Batch (size = 4)

CNN

Batch #1

Learning Rate
Architecture
Optimizer

Validation Dataset

**Batch #1**

<table>
<thead>
<tr>
<th></th>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td>0.25</td>
</tr>
<tr>
<td>2</td>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td>0.62</td>
</tr>
<tr>
<td>3</td>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td>1.12</td>
</tr>
<tr>
<td>4</td>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>

**Binary cross-entropy error** = 1.12

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Dataset

Batch (size = 4)

CNN

Learning Rate

Architecture

Optimizer

Validation Dataset

Prediction

Batch #1

Binary cross-entropy error

\[
\text{error} = - (1 \times \log(0.38) + (1 - 1) \times \log(1 - 0.38))
\]

<table>
<thead>
<tr>
<th></th>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td>0.62</td>
<td></td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td>1.12</td>
<td></td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Dataset

Batch (size = 4)

CNN

Learning Rate
Architecture
Optimizer

Batch #1

Prediction

Binary cross-entropy error

$= -(\log(0.38))$

$\mathbf{p} = 0.38 \quad \mathbf{y} = 1$

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>Dog</td>
<td>True</td>
<td>Error</td>
</tr>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td>0.25</td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td>0.62</td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td>1.12</td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Dataset

Batch (size = 4)

CNN

Validation Dataset

Batch #1

Learning Rate
Architecture
Optimizer

\[ p = 0.38 \quad y = 1 \]

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>Dog</td>
<td>True</td>
<td>Error</td>
</tr>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td>0.25</td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td>0.62</td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td>1.12</td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td></td>
</tr>
</tbody>
</table>

**Binary cross-entropy error**

\[ = 1.4 \]

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Binary cross-entropy error = 1.4

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td>0.25</td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td>0.62</td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td>1.12</td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td>1.4</td>
</tr>
</tbody>
</table>

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
batch #1

Dataset

Batch (size = 4)

Learning Rate
Architecture
Optimizer

CNN

prediction

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>Dog</td>
<td>0.84</td>
<td>0.16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cat</td>
<td>0.25</td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td>0.62</td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td>1.12</td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td>1.4</td>
</tr>
</tbody>
</table>

**Total** 3.39
Batch #1

Dataset

Batch (size = 4)

CNN

Learning Rate
Architecture
Optimizer

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>Dog</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td>0.25</td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td>0.62</td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td>1.12</td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td>1.4</td>
</tr>
</tbody>
</table>

Average total 3.39/4

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Batch #1

Dataset

CNN

Backpropagation

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td>0.25</td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td>0.62</td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td>1.12</td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td>1.4</td>
</tr>
</tbody>
</table>

Average total: 0.84

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
## Dataset

Batch (size = 4)

## Batch #1

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td>0.25</td>
</tr>
<tr>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td>0.62</td>
</tr>
<tr>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td>1.12</td>
</tr>
<tr>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td>1.4</td>
</tr>
</tbody>
</table>

Average total: 0.84

## Training error

<table>
<thead>
<tr>
<th>Batch #1</th>
<th>Batch #2</th>
<th>Batch #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Dataset

Batch (size = 4)

CNN

Learning Rate
Architecture
Optimizer

Table:

<table>
<thead>
<tr>
<th></th>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch #1</td>
<td>0.84</td>
<td>0.16</td>
<td>Cat</td>
<td>0.25</td>
</tr>
<tr>
<td>Batch #2</td>
<td>0.65</td>
<td>0.35</td>
<td>Cat</td>
<td>0.62</td>
</tr>
<tr>
<td>Batch #3</td>
<td>0.54</td>
<td>0.46</td>
<td>Dog</td>
<td>1.12</td>
</tr>
<tr>
<td>Average total</td>
<td>0.38</td>
<td>0.62</td>
<td>Cat</td>
<td>1.4</td>
</tr>
</tbody>
</table>

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
CNN Dataset

Dataset

Validation

Batch #1

Batch (size = 4)

Learning Rate

Architecture

Optimizer

prediction

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
</table>

Training error

<table>
<thead>
<tr>
<th>Batch #1</th>
<th>Batch #2</th>
<th>Batch #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Dataset

**CNN**

**Validation**

**Dataset**

**Batch** (size = 4)

**Learning Rate**

**Architecture**

**Optimizer**

**CNN**

**Prediction**

**Training error**

<table>
<thead>
<tr>
<th>Batch #1</th>
<th>Batch #2</th>
<th>Batch #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

**Average total** 0.82

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.33</td>
<td>0.67</td>
<td>Dog</td>
<td>0.58</td>
</tr>
<tr>
<td>0.23</td>
<td>0.77</td>
<td>Dog</td>
<td>0.38</td>
</tr>
<tr>
<td>0.48</td>
<td>0.52</td>
<td>Cat</td>
<td>0.97</td>
</tr>
<tr>
<td>0.61</td>
<td>0.39</td>
<td>Dog</td>
<td>1.35</td>
</tr>
</tbody>
</table>

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
CNN

Dataset

Validation

Batch #1

Prediction

Batch #2

Learning Rate

Architecture

Optimizer

Cat

Dog

True

Error

0.33

0.67

Dog

0.58

0.23

0.77

Dog

0.38

0.48

0.52

Cat

0.97

0.61

0.39

Dog

1.35

Average total

0.82

Training error

Batch #1

Batch #2

Batch #3

0.84

?

?

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Dataset

Batch (size = 4)

Learning Rate

Architecture

Optimizer

Validation

CNN

Prediction

<table>
<thead>
<tr>
<th>Batch #1</th>
<th>Batch #2</th>
<th>Batch #3</th>
<th>Average total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.82</td>
<td></td>
<td>0.82</td>
</tr>
</tbody>
</table>

Batch #1

Learning Rate

Architecture

Optimizer

Validation

CNN

Prediction

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.33</td>
<td>0.67</td>
<td>Dog</td>
<td>0.58</td>
</tr>
<tr>
<td>0.23</td>
<td>0.77</td>
<td>Dog</td>
<td>0.38</td>
</tr>
<tr>
<td>0.48</td>
<td>0.52</td>
<td>Cat</td>
<td>0.97</td>
</tr>
<tr>
<td>0.61</td>
<td>0.39</td>
<td>Dog</td>
<td>1.35</td>
</tr>
</tbody>
</table>

Training error

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
batch #1  batch #2  batch #3

<table>
<thead>
<tr>
<th>Batch #1</th>
<th>Batch #2</th>
<th>Batch #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.82</td>
<td>?</td>
</tr>
</tbody>
</table>
CNN Dataset

Dataset

Validation

Batch (size = 4)

Learning Rate

Architecture

Optimizer

CNN

Prediction

Training error

<table>
<thead>
<tr>
<th>Batch #1</th>
<th>Batch #2</th>
<th>Batch #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.82</td>
<td>?</td>
</tr>
</tbody>
</table>

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
Dataset

Batch (size = 4)

CNN

Learning Rate
Architecture
Optimizer

Batch #1
Batch #2
Batch #3

<table>
<thead>
<tr>
<th>Batch</th>
<th>0.84</th>
<th>0.82</th>
<th>?</th>
</tr>
</thead>
</table>

Training error

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.9</td>
<td>Dog</td>
<td>0.15</td>
</tr>
<tr>
<td>0.87</td>
<td>0.13</td>
<td>Cat</td>
<td>0.2</td>
</tr>
<tr>
<td>0.75</td>
<td>0.25</td>
<td>Cat</td>
<td>0.41</td>
</tr>
<tr>
<td>0.48</td>
<td>0.52</td>
<td>Dog</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Average total 0.42

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
CNN Dataset Batch (size = 4) Validation

Batch #1  Batch #2  Batch #3

0.84  0.82  0.42

Training error

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
CNN Dataset Batch (size = 4) Validation Dataset

Batch #1 Batch #2 Batch #3

Batch (size = 4) CNN

Learning Rate Architecture Optimizer

Prediction

Training error

<table>
<thead>
<tr>
<th></th>
<th>Batch #1</th>
<th>Batch #2</th>
<th>Batch #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.82</td>
<td>0.42</td>
<td></td>
</tr>
</tbody>
</table>

Calculate average training error (loss)

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
### Training error

<table>
<thead>
<tr>
<th>Batch #1</th>
<th>Batch #2</th>
<th>Batch #3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.82</td>
<td>0.42</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Calculate **average training error** (loss)

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
CNN Dataset

Batch (
size = 4
)

Learning Rate Architecture Optimizer

CNN

Prediction

Validation phase!

Batch #1

Batch #2

Batch #3

Average

Training error

<table>
<thead>
<tr>
<th>Batch #1</th>
<th>Batch #2</th>
<th>Batch #3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.82</td>
<td>0.42</td>
<td>0.69</td>
</tr>
</tbody>
</table>

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
CNN
Dataset
batch #1
batch #2
batch #3
Validation
Batch (size = 4)
Validation phase!
Learning Rate
Architecture
Optimizer
prediction
Cat
Dog
True
Error
Training error
<table>
<thead>
<tr>
<th>Batch #1</th>
<th>Batch #2</th>
<th>Batch #3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.82</td>
<td>0.42</td>
<td>0.69</td>
</tr>
</tbody>
</table>
Training error

<table>
<thead>
<tr>
<th>Batch #1</th>
<th>Batch #2</th>
<th>Batch #3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.82</td>
<td>0.42</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Validation phase!

<table>
<thead>
<tr>
<th></th>
<th>Cat</th>
<th>Dog</th>
<th>True</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch #1</td>
<td>0.9</td>
<td>0.1</td>
<td>Cat</td>
<td>0.15</td>
</tr>
<tr>
<td>batch #2</td>
<td>0.87</td>
<td>0.13</td>
<td>Cat</td>
<td>0.2</td>
</tr>
<tr>
<td>batch #3</td>
<td>0.25</td>
<td>0.75</td>
<td>Dog</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td>0.52</td>
<td>0.48</td>
<td>Cat</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Average total 0.42
Our validation error (also referred to as loss) is 0.42
Dataset

Batch #1  Batch #2  Batch #3

Validation phase!

Batch (size = 4)

CNN

Learning Rate  Architecture  Optimizer

prediction

Batch #1  Batch #2  Batch #3  Average

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.82</td>
<td>0.42</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Our validation error (also referred to as loss) is 0.42
Batch #1
During **one epoch** we train through the **entire training set**.
Batch #1
Batch #1  Batch #2
Batch #1  Batch #2  Batch #3  Validation
Epoch #1

Aver. training loss = 0.67
Validation loss = 0.42
Batch #1  Batch #2  Batch #3  Validation

Epoch #1

Aver. training loss = 0.67
Validation loss = 0.42

Epoch #2

Aver. training loss = 0.54
Validation loss = 0.41
Aver. training loss = 0.67
Validation loss = 0.42

Aver. training loss = 0.54
Validation loss = 0.41

Aver. training loss = 0.33
Validation loss = 0.37
Epoch #1
Aver. training loss = 0.67
Validation loss = 0.42

Epoch #2
Aver. training loss = 0.54
Validation loss = 0.41

Epoch #3
Aver. training loss = 0.33
Validation loss = 0.37
Epoch #1
Aver. training loss = 0.67
Validation loss = 0.42

Epoch #2
Aver. training loss = 0.54
Validation loss = 0.41

Epoch #3
Aver. training loss = 0.33
Validation loss = 0.37

Epoch #4
Aver. training loss = 0.17
Validation loss = 0.31

Epoch #5
Aver. training loss = 0.09
Validation loss = 0.29
Epoch #1
Aver. training loss = 0.67
Validation loss = 0.42

Epoch #2
Aver. training loss = 0.54
Validation loss = 0.41

Epoch #3
Aver. training loss = 0.33
Validation loss = 0.37

Epoch #4
Aver. training loss = 0.17
Validation loss = 0.31

Epoch #5
Aver. training loss = 0.09
Validation loss = 0.29
<table>
<thead>
<tr>
<th>Epoch</th>
<th>Aver. training loss</th>
<th>Validation loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>0.67</td>
<td>0.60</td>
</tr>
<tr>
<td>#2</td>
<td>0.54</td>
<td>0.41</td>
</tr>
<tr>
<td>#3</td>
<td>0.33</td>
<td>0.30</td>
</tr>
<tr>
<td>#4</td>
<td>0.17</td>
<td>0.19</td>
</tr>
<tr>
<td>#5</td>
<td>0.09</td>
<td>0.14</td>
</tr>
</tbody>
</table>
Training seems to be normal

**Epoch #1**
- Aver. training loss = 0.67
- Validation loss = 0.60

**Epoch #2**
- Aver. training loss = 0.54
- Validation loss = 0.41

**Epoch #3**
- Aver. training loss = 0.33
- Validation loss = 0.30

**Epoch #4**
- Aver. training loss = 0.17
- Validation loss = 0.19

**Epoch #5**
- Aver. training loss = 0.09
- Validation loss = 0.14
Epoch #2

Aver. training loss = 0.54
Validation loss = 0.41
Image Kernels

Explained Visually

http://setosa.io/ev/image-kernels/
Deep Convolutional Network in the spread sheet

![Spreadsheet Image](https://goo.gl/yKnwLu)
Teach a machine using your camera, live in the browser. No coding required.

Let's Go!
Training Neural Networks
(part I)

https://phiresky.github.io/neural-network-demo/
Training Neural Networks
(part II)

http://playground.tensorflow.org/
Convolutional Neural Networks
Demo

http://scs.ryerson.ca/~aharley/vis/conv/
Significant part of the lecture has been adapted from:

How Convolutional Neural Networks Work
Resources:

**Brandon Rohrer’s** youtube video: How Deep Neural Networks Work (https://youtu.be/ILsA4nyG7I0)

**Brandon Rohrer’s** youtube video: How Convolutional Neural Networks work (https://youtu.be/FmpDIaiM1eA)

Stanford University **CS231n** Convolutional Neural Networks for Visual Recognition (github.io version): http://cs231n.github.io/

**Matt Mazur’s**: A Step by Step Backpropagation Example (https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/)

**Andrej Karpathy’s blog post**: Yes you should understand backprop (https://medium.com/@karpathy/yes-you-should-understand-backprop-e2f06eab496b)

**Raul Vicente’s lecture**: From brain to Deep Learning and back (https://www.uttv.ee/naita?id=23585)

Kahoot!
That's all Folks!