3 Features

Shuffle data

Create batches

8000 Samples

8000 Samples with shuffled order

X

In case there is an order in samples. Shuffling avoids creating batches with distinct set of data.

10 batches each of size 800

Value of variable X1 for one sample

x_1

w_1

pairwise connections

Value of variable X2 for one sample

x_2

$w2$

Artifical neuron in a hidden layer

$\sigma(w_1 x_1 + w_2 x_2 + b)$

σ can be any function, even linear. It is genrally differentiable.

$f(u) = \max(0, u)$

A Rectified linear unit (ReLU) activation function. The value if either 0 or equal to to the input.

Input layer

Value of value for first variable (Size of vector = batch size)

$\bar{x}_1$

w_1

pairwise connections

Value of value for second variable (Size of vector = batch size)

$\bar{x}_2$

$w2$

Artifical neuron in a hidden layer

$\sigma(w_1 \bar{x}_1 + w_2 \bar{x}_2 + b)$

σ can be any function, even linear. It is genrally differentiable.

$f(u) = \max(0, u)$

A Rectified linear unit (ReLU) activation function. The value if either 0 or equal to to the input.

Input layer

One step

80 samples

$\bar{x}_1$ $\bar{x}_2$ $\bar{x}_3$

One batch

$\bar{x}_1$

$\bar{x}_2$

$\bar{x}_3$

Input layer with one batch of size 10

Connections between each pair

First hidden layer

Connections between each pair

Second hidden layer

Connections between each pair

Third hidden layer

Connections between each pair

Fourth hidden layer

Connections between each pair

Fifth hidden layer

Connections between each pair

$\vec{x'}_1$

$\vec{x'}_2$

$\vec{x'}_3$

Output layer

3 features

$\vec{x'}_1$ $\vec{x'}_2$ $\vec{x'}_3$

80 samples

Reconstructed batch

One forward pass

Compute loss

$$\frac{(\bar{x}_1 - \bar{x'}_1)^2 + (\bar{x}_2 - \bar{x'}_2)^2 + (\bar{x}_3 - \bar{x'}_3)^2}{10}$$

Mean squared error for the batch. Can be any function (generally, differentiable)

Update network weights

Apply backpropogation with gradient descent and update weights (Frameworks to do automatic differentiation)

Backpropogation of loss

Encoder

Decoder

Bottleneck

(Also refered to as bottleneck code or simply, code)

80 samples

x_1 x_2 x_3

One batch

$\bar{x}_1$

$\bar{x}_2$

$\bar{x}_3$

Input layer with one batch of size 10

Connections between each pair

First hidden layer

Connections between each pair

Second hidden layer

Connections between each pair

Third hidden layer

Connections between each pair

Fourth hidden layer

Connections between each pair

Fifth hidden layer

Connections between each pair

$\vec{x'}_1$

$\vec{x'}_2$

$\vec{x'}_3$

Output layer

3 features

$\vec{x'}_1$ $\vec{x'}_2$ $\vec{x'}_3$

80 samples

Reconstructed batch (Also refered to as autoencoded input)