**Week 4 – Notes**

**Deep L-layer Neural Network**

We count the layers starting with the first one (the input layer), which is 0

Close-up of a white paper with blue writing

Description automatically generated with low confidence

**Forward Propagation in a Deep Network**

Regardless of how many layers we have, the equations for any layer are the following ones:

A picture containing handwriting, font, whiteboard, typography

Description automatically generated

Even when we want to vectorize these equations, we only replace z and a with Z and A

**Getting your Matrix Dimensions Right**

**For any layer, the dimensions are the following ones:**

**A picture containing text, handwriting, font, whiteboard

Description automatically generated + the derivatives of w and b have the same dimensions**

Additionally, z[l] and a[l] have the same dimensions as b[l]

In the vectorized implementation the dimensions are:

Z[l] -> (n[l], m)

A[l] -> (n[l], m)

However, even in the vectorized implementation, W[l] and b[l] have the same dimensions

**Why Deep Representations?**

We want to use deep neural networks because they can learn complex features, as initial layers detect simpler features and deeper and deeper layers combine the previous detected features into more complex representations

For example a face detector detects: types of edges -> combine them in eyes / noses etc -> form faces

Circuit theory and deep learning: small and deep neural networks can compute the same functions as shallower and huge deep neural networks (small / huge in terms of the number of neurons) – example: computing x1 xor x2 xor x3 … xor xn

**Building Blocks of Deep Neural Networks**

During the forward propagation:

use a[l-1], w[l] and b[l] to compute a[l] and cache z[l], w[l], b[l], and a[l-1]

During the backward propagation:

Use da[l] and z[l] to compute the da[l-1], dw[l] and db[l]

A picture containing text, handwriting, diagram, child art

Description automatically generated

In the end we use dw[l] and db[l] to update w[l] and b[l]

**Forward and Backward Propagation**

Vectorized forward propagation for layer l:

A close-up of writing on a white board

Description automatically generated with low confidence

The seed is represented by X (A[0])

Vectorized backward propagation for layer l:

A whiteboard with writing on it

Description automatically generated with low confidence

During the backpropagation we compute da[0], but it’s useless because we won’t use it

The seed is represented by

[Feedforward Neural Networks in Depth, Part 1: Forward and Backward Propagations | I, Deep Learning (jonaslalin.com)](https://jonaslalin.com/2021/12/10/feedforward-neural-networks-part-1/)

[Feedforward Neural Networks in Depth, Part 3: Cost Functions | I, Deep Learning (jonaslalin.com)](https://jonaslalin.com/2021/12/22/feedforward-neural-networks-part-3/)

A picture containing font, line, white, number

Description automatically generated

A picture containing font, text, line, white

Description automatically generated

**Parameters vs Hyperparameters**

Hyperparameters are set by humans and help the network to learn the actual parameters W and b

Types of hyperparameters: learning rate alpha, number of iterations, hidden layers, hidden units, the choice of the activation functions and so on

The applied deep learning process is very empirical and you have to pick and test various hyperparameters

It’s important that even if you have a model that is trained with some hyperparameters and you have good results, just try every couple of months to re-do the search because the data / infrastructure has been changed

**What does this have to do with the brain?**

It’s not known if the brain works similar to forward and backward propagation

We only know that deep learning is a very flexible way to learn complex functions