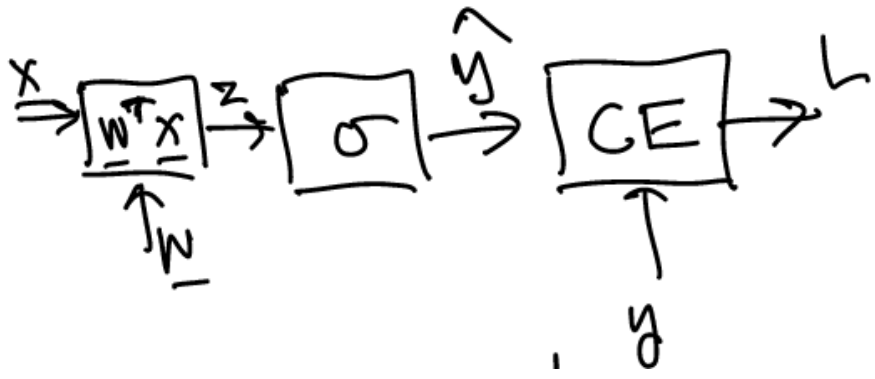
Neural Network Assignment - Answers

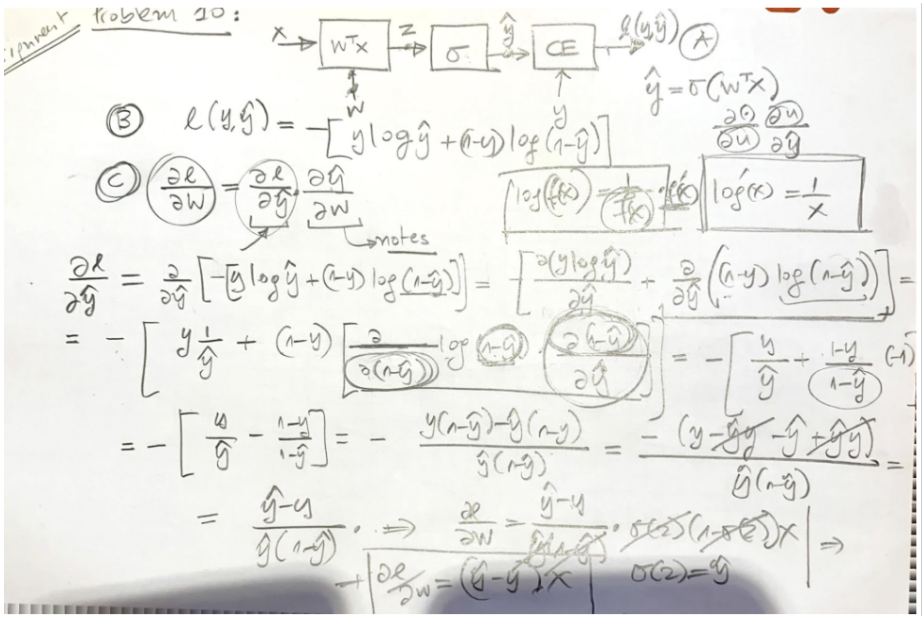
Neural Networks Assignment - Answers

Backpropagation (20 points)

$$\frac{\partial L}{\partial \mathbf{w}}$$



Answer



Please accept as correct any very close to this one derivation. Aka dont penalize if the final expression is wrong but use your judgement to award points.

Tensorflow Playground (60 points. 10 points per question)

The TensorFlow Playground is a handy neural network simulator built by the TensorFlow team. In this exercise, you will train several binary classifiers in just a few clicks, and tweak the model's architecture and its hyperparameters to gain some intuition on how neural networks work and what their hyperparameters do.

1. Try training the default neural network by clicking the Run button (top left). Notice how it quickly finds a good solution for the classification task. The neurons in the first hidden layer have learned simple patterns, while the neurons in the second hidden layer have learned to combine the simple patterns of the first hidden layer into more complex patterns. Why that is?

Answer: By definition the second layer added parameters in the hypothesis set and therefore the network is able to create more complex representations.

2. Activation functions. Try replacing the tanh activation function with a ReLU activation function, and train the network again. Does it find the solution faster or slower? Why is that?

Answer: It finds a solution even faster, but this time the boundaries are linear. This is due to the shape of the ReLU function.

3. The risk of local minima. Modify the network architecture to have just one hidden layer with three neurons. Train it multiple times (to reset the network weights, click the Reset button next to the Play button). Why the training time has a wide variability?

Answer: Because the network can get stuck in local minima.

3. Remove one neuron to keep just two. Notice that the neural network is now incapable of finding a good solution - why that is?

Answer: The model has too few parameters and systematically underfits the training set.

4. Set the number of neurons to eight, and train the network several times. Has the training time (time to convergence) improved? Why that is?

Answer: Large neural networks almost never get stuck in local minima, and even when they do these local optima are almost as good as the global optimum. However, they can still get stuck on long plateaus for a long time.

5. Select the spiral dataset (the bottom-right dataset under "DATA"), and change the network architecture to have four hidden layers with eight neurons each. Notice that training takes much longer and often gets stuck on plateaus for long periods of time. Also notice that the neurons in the highest layers (on the right) tend to evolve faster than the neurons in the lowest layers (on the left). Can you explain how this may be related to gradient flow through the network?

Answer: This problem, called the "vanishing gradients" problem, can be alleviated with better weight initialization and other techniques such as better better optimizers.

Tensorflow API (20 points)

1. (5 points) Submit your notebook URL that allows the notebook here to be executed.

Answer: You should check that the notebook runs.

2. (15 points) Start reducing the number of cats in the dataset and plot the accuracy of the predicting the cat class as the population of cats becomes 90%, 70%, 50%, 30%, 10% of the original. For each population size present the hyperparameter optimized result using AutoKeras. Explain your findings.

Answer: See pfd file of the notebook by a student that explains well what is going on.