MA 751
M. Kon

Problem Set 5
Due Friday, 3/4/22

Note that the coming week we will have a Data Assignment due on Tuesday. For this reason, this problem set will be due a day later than usual, on Friday of next week.

Lectures 9, 10

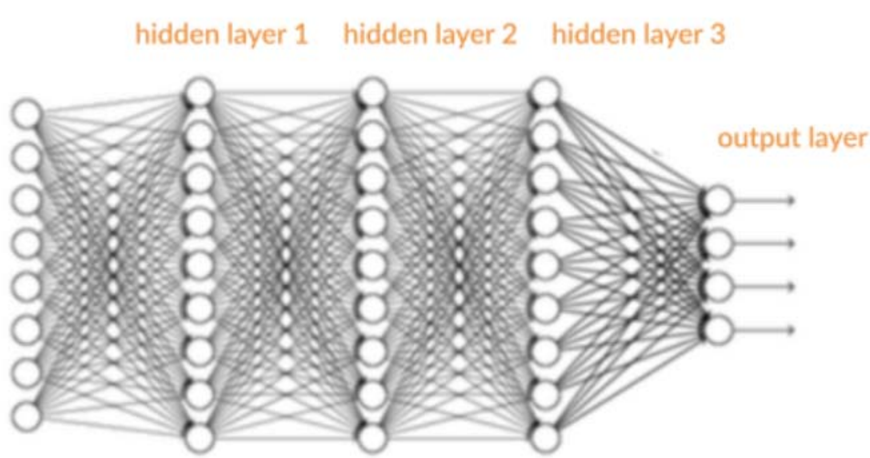
We have gotten a detailed look at backpropagation in the textbook, an idea that extends also to networks with any number of layers. Essentially, backprop is an extremely organized way to do gradient descent, by segregating it into components corresponding to separate (batches of) examples, and also gradient descending the parameeters of one layer at a time.

Deep neural nets have a lot of deep ideas behind them, some of which are possible to state precisely, and some of which are just being developed. Empirically, deep networks work demonstrably well both in image recognition (as in their performance on Imagenet) and in biological organisms (from insects to humans), which also (at least seem to) use them with good results.

Reading: 11.1-11.8, class material

Problems:

1. Given a ReLU neural network discussed in class, consider a 'typical' architecture



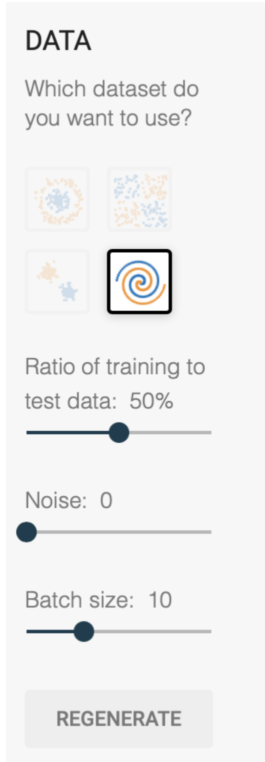
Notice that there are additional options at each node, to perform additional operations besides just the operation $F_k(\boldsymbol{x}) = \mathrm{ReLU}(A_k\boldsymbol{x} + \boldsymbol{b})$. For example, we have discussed pooling, in which we can place an additional operation before we define the kth layer, obtained by taking the vector $F_k(\boldsymbol{x})$ and performing a pooling operation, for example taking the maximum of each even-odd pair of successive entries in $F_k(\boldsymbol{x})$, and replacing them by a single entry forming the maximum of the two (and thus obtaining a vector of half the length). There can also be plenty of compositions of functions of the form $F_k(\boldsymbol{x})$. Answer the following questions in the form of theorems.

If $F_1(\boldsymbol{x}),\ F_2(\boldsymbol{x})$ are continuous piecewise linear functions, what about:

(a) $G(\boldsymbol{x}) \equiv \max(F_1(\boldsymbol{x}), F_2(\boldsymbol{x}))$?
(b) $H(\boldsymbol{x}) \equiv F_1(\boldsymbol{x}) + F_2(\boldsymbol{x})$?
(c) $I(x) \equiv F_2 \circ F_1(\boldsymbol{x})$?

2. This problem refers to the toolkit at playground.tensorflow.org. Consider the example of separating the blue ball from the orange ring, using the ReLU activation function (do not use the alternative tanh activation function). The goal is to build a network which can separate the two regions (with a white polygon).
(a) See if you can succeed with $K = 4$ neurons in the middle layer. How many linear pieces does $F(\boldsymbol{x})$ have? How many sides are there in the polygon?
(b) What occurs when $K = 3$?
(c) How about $K = 2$? How many folds are there in the computed function F? How many sides in the piecewise linear separator?

3. Consider the fourth example at the tensorflow web site. This is a spiral with alternating blue and orange arms. You will find it on the left margin of the entry page to playground.tensorflow.org. This is partly an exploratory problem.



(a) Try to classify this with one hidden layer (so that the network has three layers with one output neuron).
(b) What does Funahashi's theorem say about whether this can be done?
(c) Show the results as the number q of neurons in the hidden layer.increases. How does the quality of the classification change?
(d) Now try with two hidden layers. For the same numbers of neurons, which of the two cases (a single or two hidden layers) seem to do the job better?
(e) What about 3 layers?

4. (a) Consider 20 fold lines in the plane. What is the maximum number of linear pieces that a function can have with such a set of fold lines.
(b) Consider the case where the fold lines consist of 10 vertical and 10 horizontal lines. What is the answer in this case?
(c) What about the case of k vertical and $20 - k$ horizontal for $k \leq 20$?