Experiment# 3

Artificial Neural Network EE-3064

U. B. Mansoor

Tuesday, April 13th, 2021

Lab 3 on CoLab

Objective

By the end of this lab, students should be able to:

1. Use Playground.tensorflow.org web based application to analyze a Neural Network.
2. Build a NN using Playground.tensorflow.org to learn nonlinear models.

Background

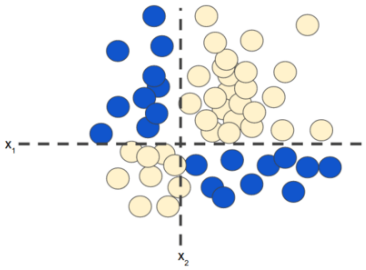

Figure 1: Nonlinear Classification Problem.

"Nonlinear" means that you can't accurately predict a label with a model of the form $b + w_1x_1 + w_2x_2$. In other words, the "decision surface" is not a line.

A First Neural Network

In this exercise, we will train our first little neural net using the web app located at the Playground.tensorflow.org website. Neural nets will give us a way to learn nonlinear models without the use of explicit feature crosses. The classification problem is known as the XOR problem as depicted in Figure 1.

Task 1: The model as given combines our two input features into a single neuron. Will this model learn any nonlinearities? Run it to confirm your guess.

Task 2: Try increasing the number of neurons in the hidden layer from 1 to 2, and also try changing from a Linear activation to a nonlinear activation like ReLU. Can you create a model that can learn nonlinearities? Can it model the data effectively?

Task 3: Try increasing the number of neurons in the hidden layer from 2 to 3, using a nonlinear activation like ReLU. Can it model the data effectively? How model quality vary from run to run?

Task 4: Continue experimenting by adding or removing hidden layers and neurons per layer. Also feel free to change learning rates, regularization, and other learning settings. What is the smallest number of neurons and layers you can use that gives test loss of 0.177 or lower?

Does increasing the model size improve the fit, or how quickly it converges? Does this change how often it converges to a good model? For example, try the following architecture:

- First hidden layer with 3 neurons.
- Second hidden layer with 3 neurons.
- Third hidden layer with 2 neurons.

S.#	Acivation Function	# of Hidden Layers	# of Neurons	Epochs	Training Error	Test Error
1	Linear/ ReLU	0	1	500	0.491	0.510
2	ReLU	1	2	500	0.172	0.307
3	ReLU	1	3	500	0.206	0.254
4	ReLU	1	4	500	0.04	0.06
5	ReLU	2	6	500	0.01	0.02

Table 1: XOR classification problem with 50% training data, 35% noise and a batch size of 10

LabWork

Neural Net Spiral

This data set is a noisy spiral. Obviously, a linear model will fail here, but even manually defined feature crosses may be hard to construct.

Task 1: Train the best model you can, using just X1 and X2. Feel free to add or remove layers and neurons, change learning settings like learning rate, regularization rate, and batch size. What is the best test loss you can get? How smooth is the model output surface?

Task 2: Even with Neural Nets, some amount of feature engineering is often needed to achieve best performance. Try adding in additional cross product features or other transformations like sin(X1) and sin(X2). Do you get a better model? Is the model output surface any smoother?

S.#	Acivation Function	# of Hidden Layers	# of Neurons	Epochs	Training Error	Test Error
1	ReLU	1	4	1000	0.471	0.472
2	ReLU	2	8	1000	0.312	0.348
3	ReLU	3	12	1000	0.196	0.279
4	ReLU	4	12	1000	0.071	0.106
5	ReLU	5	15	1000	0.176	0.196
6	ReLU	6	22	1000	0.015	0.038
7	ReLU	4	12	1000	0.188	0.290

Table 1: Spiral data classification problem with 50% training data, 80% noise and a batch size of 10