

9.28.22

Wednesday, September 28, 2022 5:38 PM

Nathan has the NN pipeline up and running. (if you press enter, cells execute without error.)

This is code from the old project at UCSC where he hand-wrote a fully-trainable NN.

This is usefull because we can see into the "guts" of the network, but difficult to make high-level changes (eg: the topology of the network)

We will shift directions and start using tensorflow

High-level guiding principles:

1. Use tensorflow
2. Be able to change the input data 'easily' (only change a single line or variable). Sometimes we will want to use p_x, p_y, p_z, E as input, but sometimes it might be better to use $(p_x^2, p_y^2, p_z^2, E^2)$ as input to the network.
3. Be able to change the output of the network 'easily'. Sometimes we want to learn m_W , and sometimes we might want to learn m_W^2
4. Be able to change the network topology 'easily.'

Network Training Goals

1. Use the 4-momentum of the W ($PID = -24$) to learn $m_W^2 = (E^2 - p^2)_W$
2. Use the 4-momentum of the lepton ($PID = -11$) to learn $m_W^2 = (E^2 - p^2)_L$

Output Goals

1. Output target
2. Error per epoch
3. Access to the weights (not necessarily to output these, but to know how to access them if we need them)