

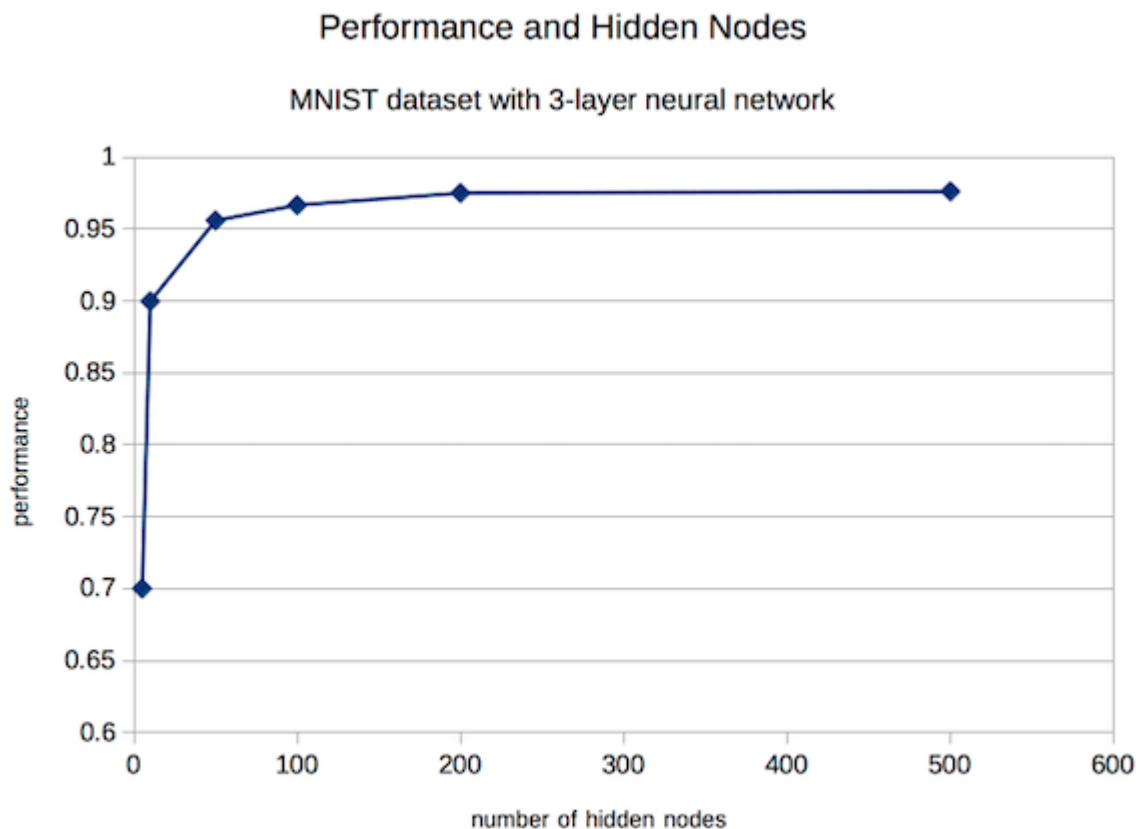
Change Network Shape

How changing the network shape can bring improvement in the accuracy of result? Let's find out.

One thing we haven't yet tried, and perhaps should have earlier, is to change the shape of the neural network. Let's try changing the number of middle hidden layer nodes. We've had them set to 100 for far too long! Before we jump in and run experiments with different numbers of hidden nodes, let's think about what might happen if we do. The hidden layer is the layer which is where the learning happens. Remember the input nodes simply bring in the input signals, and the output nodes simply push out the network's answer. It's the hidden layer (or layers) which have to learn to turn the input into the answer. It's where the learning happens. Actually, it's the link weights before and after the hidden nodes that do the learning, but you know what I mean.

If we had too few hidden nodes, say 3, you can imagine there is no way there is enough space to learn whatever a network learns, to somehow turn all the inputs into the correct outputs. It would be like asking a car with 5 seats to carry 10 people. You just can't fit that much stuff inside. Computer scientists call this kind of limit a *learning capacity*. You can't learn more than the learning capacity, but you can change the vehicle, or the network shape, to increase the capacity.

What if we had 10000 hidden nodes? Well, we won't be short of learning capacity, but we might find it harder to train the network because now there are too many options for where the learning should go. Maybe it would take 10000s of epochs to train such a network. Let's run some experiments and see what happens.



You can see that for low numbers of hidden nodes the results are not as good for higher numbers. We expected that. But the performance from just 5 hidden nodes was 0.7001. That is pretty amazing given that from such few learning locations the network is still about 70% right. Remember we've been running with 100 hidden nodes thus far. Just 10 hidden nodes get us 0.8998 accuracy, which again is pretty impressive. That's $1/10^{th}$ of the nodes we've been used to, and the network's performance jumps to 90%. This point is worth appreciating. The neural network is able to give really good results with so few hidden nodes, or learning locations. That's a testament to their power.

As we increase the number of hidden nodes, the results do improve but not as drastically. The time taken to train the network also increases significantly, because each extra hidden node means new network links to every node in the preceding and next layers, which all require lots more calculations! So we have to choose a number of hidden nodes with a tolerable runtime. For my computer that's 200 nodes. Your computer may be faster or slower. We've also set a new record for accuracy, with 0.9751 with 200 nodes. And a long run with 500 nodes gave us 0.9762. That's really good compared to the benchmarks listed on LeCun's [website](#). Looking back at the graphs you can see

that the previous stubborn limit of about 95% accuracy was broken by changing the shape of the network.