# Chapter 11.b Exercises

1. What is momentum?

Similar to vanilla gradient descent except that instead of the previous gradient being ignored at each iteration, it is now kept to a certain degree (depending on your decay rate) and the new gradient is added onto it. Essentially if the gradient is continually in the same direction, it will speed up.

1. What are three examples of adaptive optimizers? Which one is the best?

AdaGrad, RMSprop, and Adam. Adam is the best empirically because it takes the best parts of RMSprop and momentum (RMSprop is just a better AdaGrad anyway).

1. What is dropout? (How does it work, and what does it achieve)

Dropout is the concept of ignoring some random neurons in a DNN each training step. This forces the neural net to become more robust by not relying on any individual neuron or neurons; the output of a neuron is forced to be useful because it’s neighbors in the layer may be ignored, and this in turn makes the input from each neuron in the previous layer useful (well, they’re the same thing really).

1. When would you use (in an output or hidden layer)
   1. SeLU

When you want to avoid the vanishing & exploding gradients problems through internal (hence, faster) self-normalization.

* 1. Leaky ReLU

When the neurons are outputting negative values (which regular ReLU would compute a 0 gradient for, making them unchanging and basically dead, whereas leaky ReLU allows small change).

* 1. Logistic

When using Batch Normalization, and so you don’t need to worry about the vanishing gradients problem. There isn’t really much of a benefit to doing so though.

* 1. Softmax

Useful for multi-classification because it produces what is essentially a probability distribution (can be used for binary as well, but sigmoid can do that for you). Primarily used in the final layer for a classifier neural net.

1. How should you initialize the weights of your neural network layer?

This depends on the activation function used (taken from class notes). Use

1. Glorot initialization for sigmoid, softmax, tanh, or none
2. He for ReLU and it’s variants
3. LeCun for SELU

(I also read [this](https://towardsdatascience.com/weight-initialization-in-neural-networks-a-journey-from-the-basics-to-kaiming-954fb9b47c79) article, which was pretty informative as to why this is)