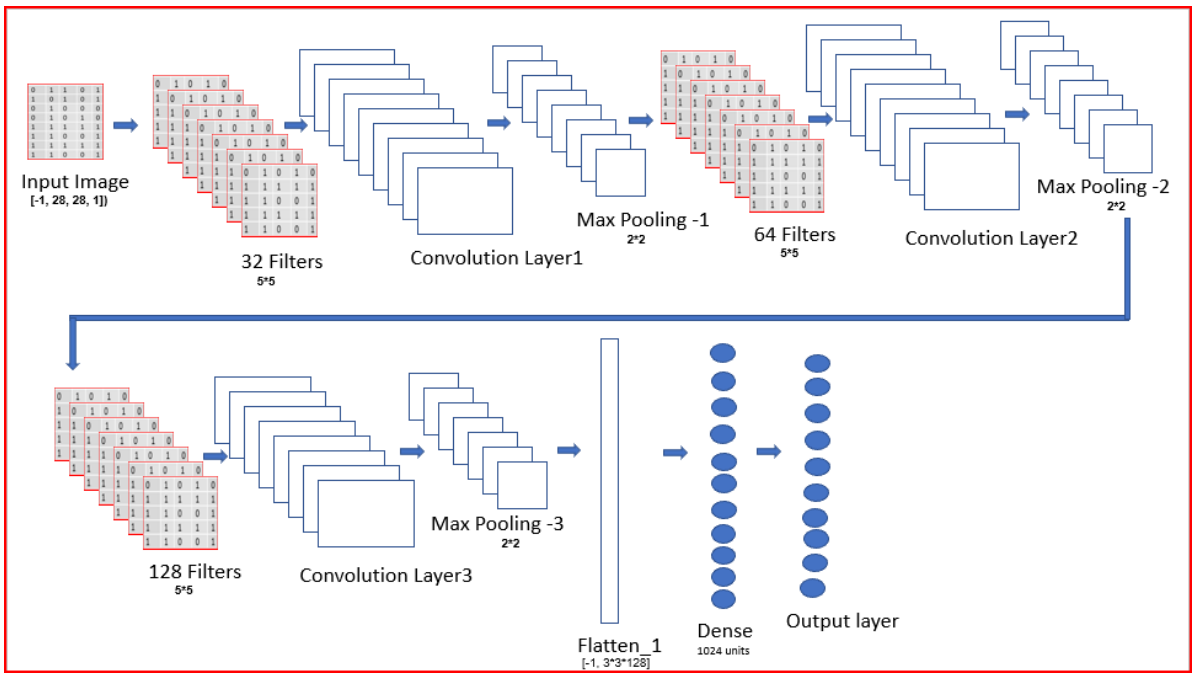
CS 6421: Programming Assignment 1

We want to explore the impact of different architectures and parameters on network predictive accuracy.

Here is an example of an architecture (this is different to the baseline, e.g., we have a dense layer of 256 instead of 1024):



Architecture of a CNN

Here is the baseline code:
https://github.com/gmprovan/CS6421-Assignment1/blob/master/Copy_of_Copy_of_Copy_of_fashion_mnist_keras.ipynb

This code has instructions for the 2 steps, which are running the experiments described below, and then coding a TensorFlow version of the model specification.

Modification to Architecture

Consider the code fragments for implementing a CNN to classify the MNIST fashion domain.

We refer to a model using the parameter set [(64,2),(32,2), (0.3,0.3)], which refers to

> [(convolution layer1 filters, convolution layer1 kernel size),
> (convolution layer2 filters, convolution layer2 kernel size),
> (convolution layer1 Dropout-probability, convolution layer2 Dropout-probability)].

Thus [(64,2),(32,2), (0.3,0.3)] is the parameter set for the baseline model---see the provided code.

Table 1: Baseline model parameters

Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	64	2x2	ReLU		
Pooling		2x2			
Dropout				0.3	
Convolution	32	2x2	ReLU		
Pooling		2x2			
Dropout				0.3	
Dense	ReLU				256
dense	softmax				10

A **modification to a parameter** means that we replace only that parameter. So in changing the baseline model:

- the filters modification (x,y) means that we run a model with parameter set [(x,2),(y,2), (0.3,0.3)].
- the kernel modification (j,k) means that we run a model with parameter set [(64,j),(32,k), (0.3,0.3)].
- the dropout modification (a,b) means that we run a model with parameter set [(64,2),(32,2), (a,b)].

1. Compile the basic model
2. Compare different architectures

I will divide the class into 4 different groups, based on last digit in your student number:

1: number ending in {0,1,2}
2: number ending in {3,4}
3: number ending in {5,6}
4: number ending in {7,8,9}

Each group will have a subset of 11 architectures to explore.

Report your results in a table, reporting predictive accuracy on the test set:

Architecture parameters *i*, accuracy *i*

Table 2: 7 experiments for each group based on single-parameter-type modification

Group	Filters	Kernel	Dropout	Extra Layer
1	(32,16)	(5,3)	(3, .25)	[(64,2),(32,2), (0.3,0.3)]
	(64,64)	(3,2)	(.25, .25)	
2	(128,64)	(5,5)	(.4,.4)	[(32,2),(32,2), (0.3,0.3)]
	(64,64)	(3,3)	(.25, .25)	
3	(32,32)	(3,3)	(.5, .25)	[(64,2),(64,2), (0.3,0.3)]
	(64,64)	(4,4)	(.25, .25)	
4	(32,32)	(3,3)	(.35, .25)	[(32,2),(16,2), (0.25,0.25)]
	(32,16)	(5,5)	(.2, .2)	[(16,2),(16,2), (0.25,0.25)]

Each group will have 4 architectures with multiple simultaneous modifications, as defined below:

Group 1

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	32	2x2	ReLU		
		Pooling		2x2			
		Dropout				0.3	
Convolution	2	Convolution	16	2x2	ReLU		
		Pooling		2x2			
		Dropout				0.3	
Decision		Dense			ReLU		256
Output		dense			softmax		10

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	32	5x5	ReLU		
		Pooling		2x2			
		Dropout				0.2	
Convolution	2	Convolution	16	5x5	ReLU		
		Pooling		2x2			
		Dropout				0.2	
Decision		Dense			ReLU		256
Output		dense			softmax		10

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	128	2x2	ReLU		
		Pooling		2x2			
		Dropout				0.3	
Convolution	2	Convolution	64	2x2	ReLU		
		Pooling		2x2			
		Dropout				0.3	
Decision		Dense			ReLU		256
Output		dense			softmax		10

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	128	3x3	ReLU		
		Pooling		3x3			
		Dropout				0.3	
Convolution	2	Convolution	64	3x3	ReLU		
		Pooling		3x3			
		Dropout				0.3	
Decision		Dense			ReLU		256
Output		dense			softmax		10

Group 2

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	32	4x4	ReLU		
		Pooling		2x2			
		Dropout				0.3	
Convolution	2	Convolution	32	4x4	ReLU		
		Pooling		2x2			
		Dropout				0.3	
Decision		Dense			ReLU		256
Output		dense			softmax		10

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	32	5x5	ReLU		
		Pooling		2x2			
		Dropout				0.2	
Convolution	2	Convolution	32	5x5	ReLU		
		Pooling		2x2			
		Dropout				0.3	
Decision		Dense			ReLU		256
Output		dense			softmax		10

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	64	2x2	ReLU		
		Pooling		2x2			
		Dropout				0.3	
Convolution	2	Convolution	64	2x2	ReLU		
		Pooling		2x2			
		Dropout				0.3	
Decision		Dense			ReLU		256
Output		dense			softmax		10

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	64	5x5	ReLU		
		Pooling		3x3			
		Dropout				0.4	
Convolution	2	Convolution	64	5x5	ReLU		
		Pooling		3x3			
		Dropout				0.3	
Decision		Dense			ReLU		256
Output		dense			softmax		10

Group 3

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	32	3x3	ReLU		
		Pooling		3x3			
		Dropout				0.3	
Convolution	2	Convolution	16	2x2	ReLU		
		Pooling		2x2			
		Dropout				0.3	
Decision		Dense	ReLU				256
Output		dense	softmax				10

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	32	5x5	ReLU		
		Pooling		3x3			
		Dropout				0.4	
Convolution	2	Convolution	16	5x5	ReLU		
		Pooling		2x2			
		Dropout				0.3	
Decision		Dense	ReLU				256
Output		dense	softmax				10

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	128	3x3	ReLU		
		Pooling		2x2			
		Dropout				0.3	
Convolution	2	Convolution	64	2x2	ReLU		
		Pooling		2x2			
		Dropout				0.25	
Decision		Dense	ReLU				256
Output		dense	softmax				10

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	128	3x3	ReLU		
		Pooling		3x3			
		Dropout				0.4	
Convolution	2	Convolution	64	3x3	ReLU		
		Pooling		3x3			
		Dropout				0.4	
Decision		Dense	ReLU				256
Output		dense	softmax				10

Group 4

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	32	4x4	ReLU		
		Pooling		2x2			
		Dropout				0.25	
Convolution	2	Convolution	32	4x4	ReLU		
		Pooling		2x2			
		Dropout				0.25	
Decision		Dense	ReLU				256
Output		dense	softmax				10

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	32	5x5	ReLU		
		Pooling		3x3			
		Dropout				0.3	
Convolution	2	Convolution	32	5x5	ReLU		
		Pooling		2x2			
		Dropout				0.3	
Decision		Dense	ReLU				256
Output		dense	softmax				10

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	64	4x4	ReLU		
		Pooling		4x4			
		Dropout				0.4	
Convolution	2	Convolution	64	2x2	ReLU		
		Pooling		2x2			
		Dropout				0.3	
Decision		Dense	ReLU				256
Output		dense	softmax				10

Type	Layer #	Type	# filters	Kernel-size	activation	Dropout probability	Size
Convolution	1	Convolution	64	5x5	ReLU		
		Pooling		3x3			
		Dropout				0.25	
Convolution	2	Convolution	64	5x5	ReLU		
		Pooling		3x3			
		Dropout				0.3	
Decision		Dense	ReLU				256
Output		dense	softmax				10

Assignment [100 marks]

Each person must turn in:

a. [40 marks] Jupyter notebook code showing the TensorFlow modifications
b. [20 marks] A table describing the results of the experiments on 11 models
c. [40 marks] A brief summary of why any significantly different results occur due to modifications to the architecture.