# CNN-Fold: Protein Fold Recognition by Deep Convolutional Neural Networks

Tyler Banks

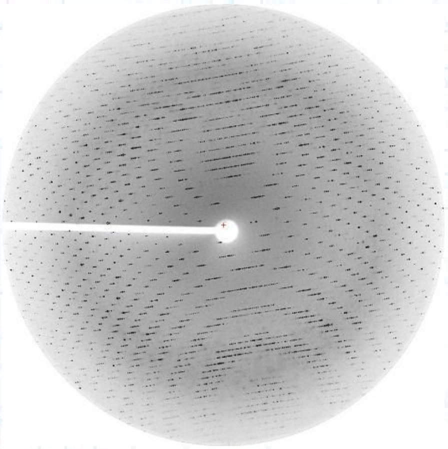
Presented to:
Dr. Jianlin Cheng, Advisor
Dr. Rohit Chadha
Dr. Jeffrey Uhlmann

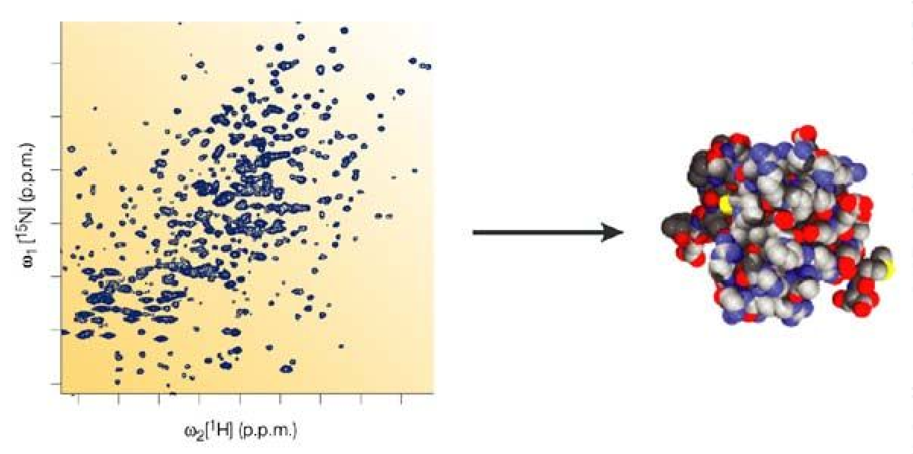
# Motivation

- Proteins structure determines function
  - Medicine
  - Biotechnology
- High discovery rate
- Known to unknown
  - 1:200

# X-Ray Crystallography and NMR Spectroscopy

- X-Ray Crystallography
  - High resolution microscopy
- NMR Spectroscopy
  - Quantum properties of the nucleus



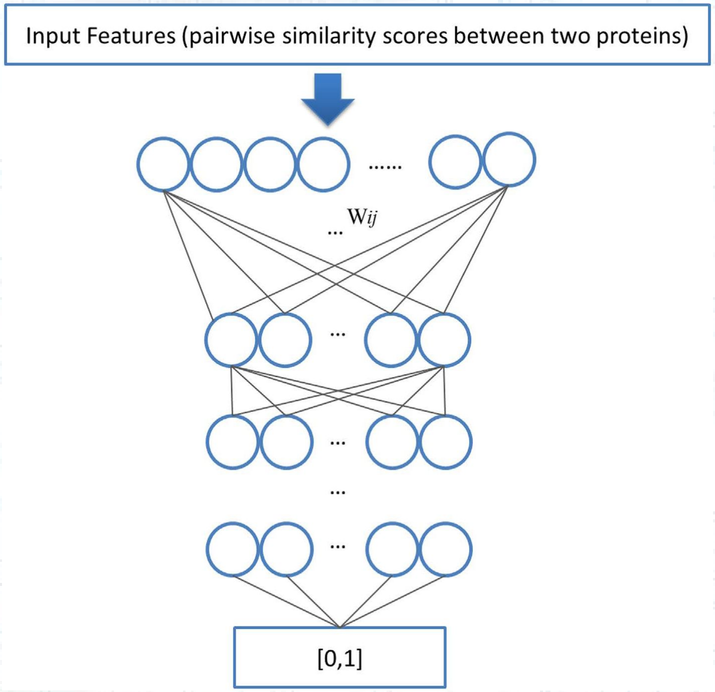


# Machine Learning Techniques

- Support Vector Machines (SVMs)
  - < 50% error rate
- Neural Networks
- Deep Learning
  - Deep Belief networks
  - 84.5% recognition rate

# DN-Fold

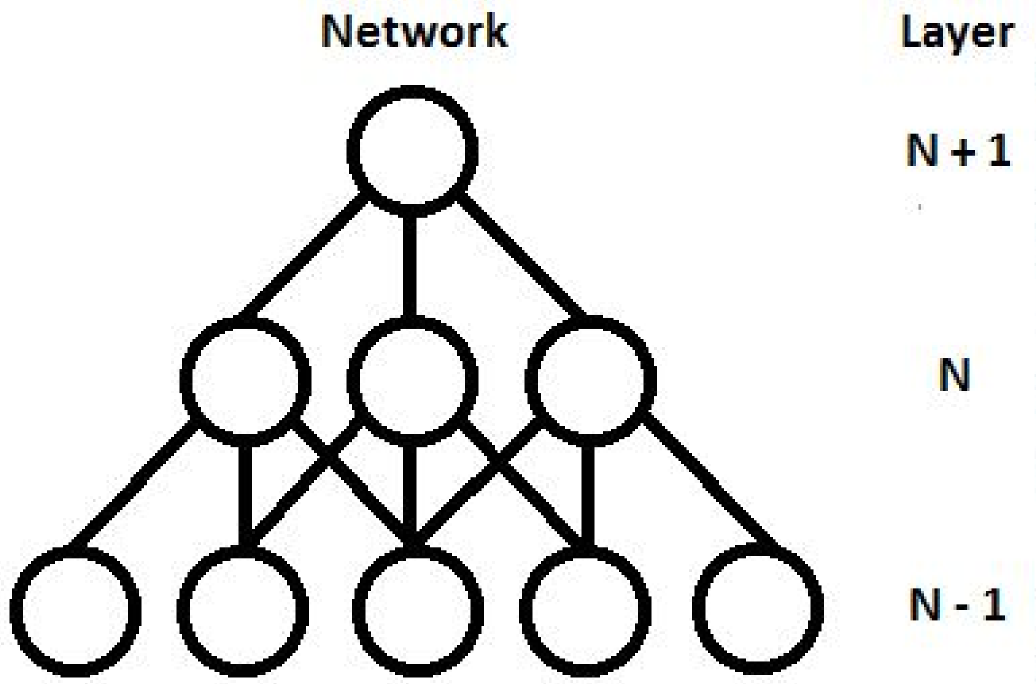
- Deep Belief Networks
  - Restricted Boltzman Machines
  - Generative Autoencoders
- Binary classification problem
  - 976 Proteins
  - (n2-n)
  - Trained and Tested


Input Features (pairwise similarity scores between two proteins)
......
Wij
...
...
...
...
...
[0,1]

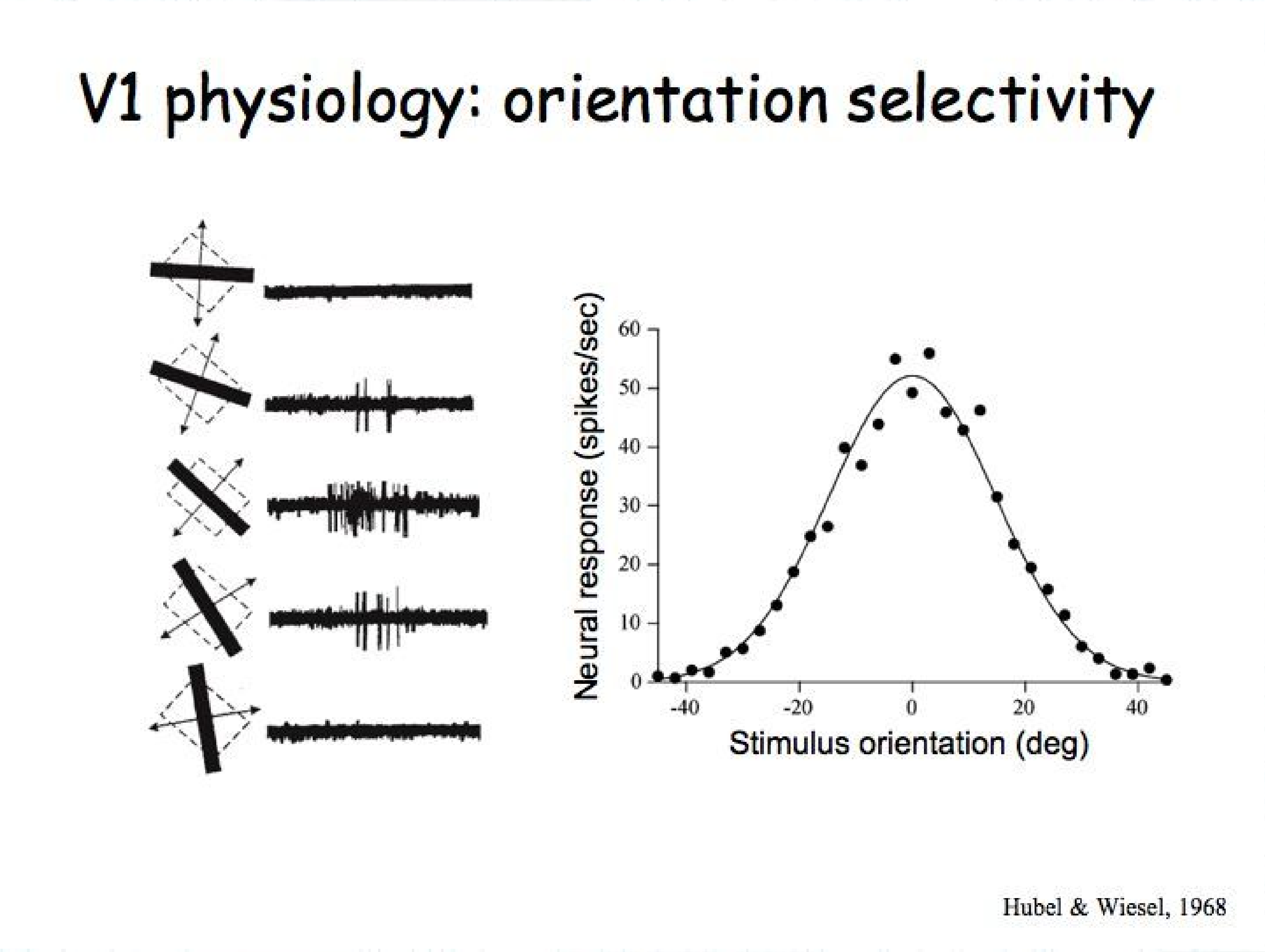
# Convolutional Neural Networks

- Receptive Fields
- Surrounding area & Hidden properties
- Fully Connected Deep Neural Network

- Images
- Sounds



# Biological Inspiration

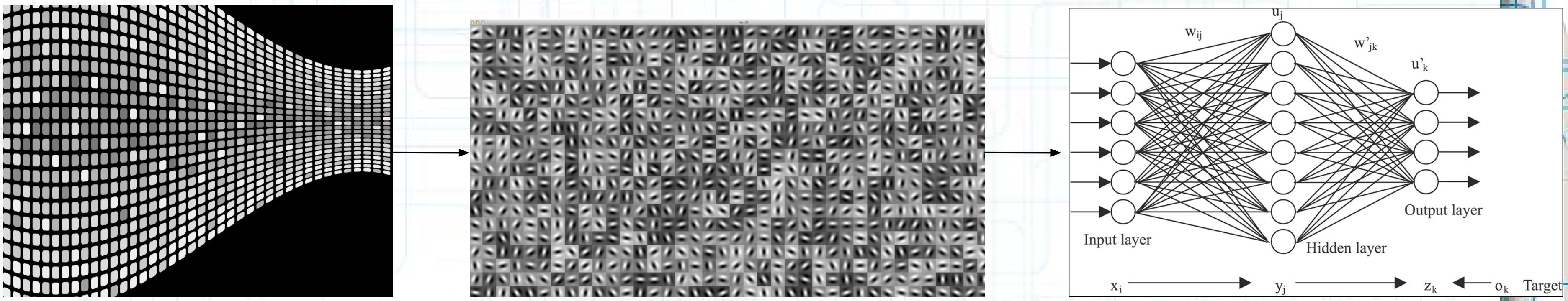


V1 physiology: orientation selectivity

Hubel & Wiesel, 1968

# Convolutional Layers

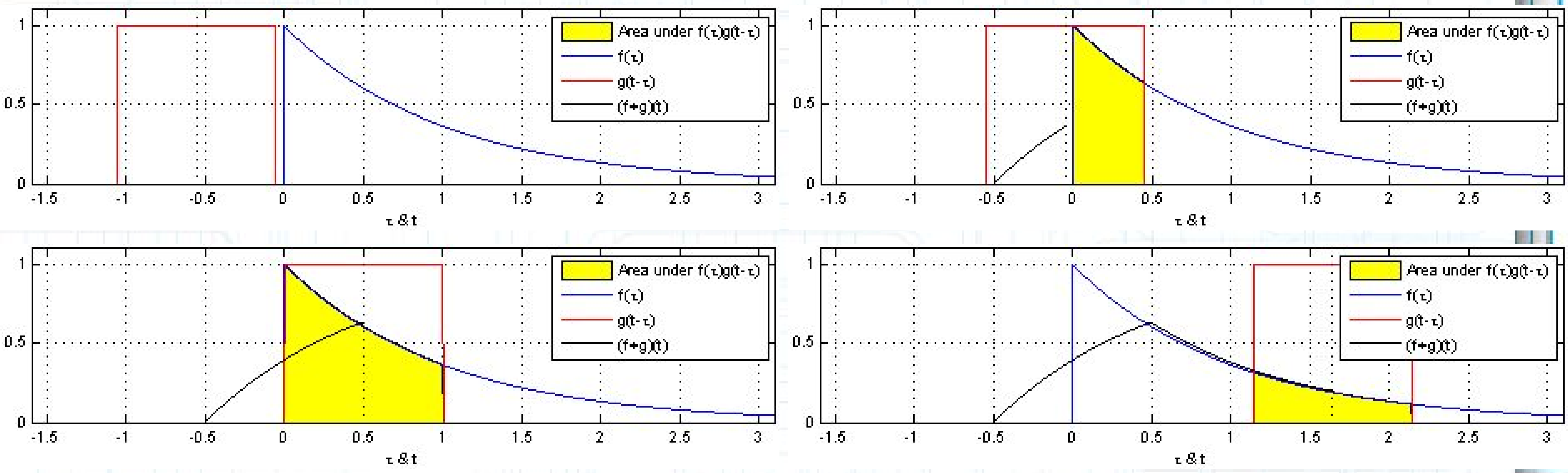
- V1 in the visual cortex
- Input → Eyes
- Filter → V1
- Output → Higher level cortical regions



# Mathematical Convolution

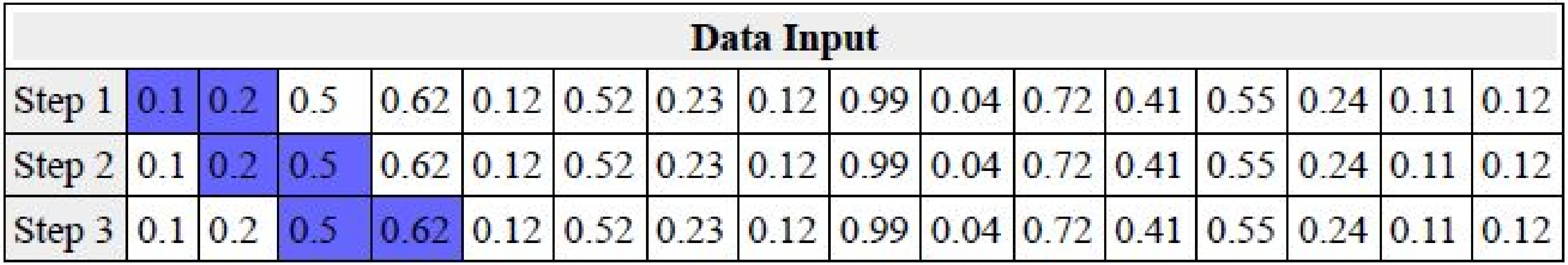
- Multiplying two function mathematically
- Produces an integral

$$\int_{-\infty}^{\infty} \delta(\tau)\, g(t-\tau)\, d\tau = g(t)$$



# Discrete Convolution

- Filters used have discrete stride lengths
- Snapshots taken



| Data Input | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Step 1 | 0.1 | 0.2 | 0.5 | 0.62 | 0.12 | 0.52 | 0.23 | 0.12 | 0.99 | 0.04 | 0.72 | 0.41 | 0.55 | 0.24 | 0.11 | 0.12 |
| Step 2 | 0.1 | 0.2 | 0.5 | 0.62 | 0.12 | 0.52 | 0.23 | 0.12 | 0.99 | 0.04 | 0.72 | 0.41 | 0.55 | 0.24 | 0.11 | 0.12 |
| Step 3 | 0.1 | 0.2 | 0.5 | 0.62 | 0.12 | 0.52 | 0.23 | 0.12 | 0.99 | 0.04 | 0.72 | 0.41 | 0.55 | 0.24 | 0.11 | 0.12 |

- Activation Map

# Producing an Activation Map

**Input (with a pad of 1) (7x7x1)**

| | | | | | | |
|---|---|---|---|---|---|---|
| **0** | **0** | **0** | 0 | 0 | 0 | 0 |
| **0** | **1** | **1** | 2 | 2 | 2 | 0 |
| **0** | **0** | **2** | 1 | 2 | 2 | 0 |
| 0 | 0 | 2 | 0 | 0 | 1 | 0 |
| 0 | 1 | 2 | 2 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Filter F0 (3x3x1)**

| | | |
|---|---|---|
| **-1** | **1** | **1** |
| **-1** | **-1** | **0** |
| **-1** | **0** | **1** |

**Bias B0 (1x1x1)**

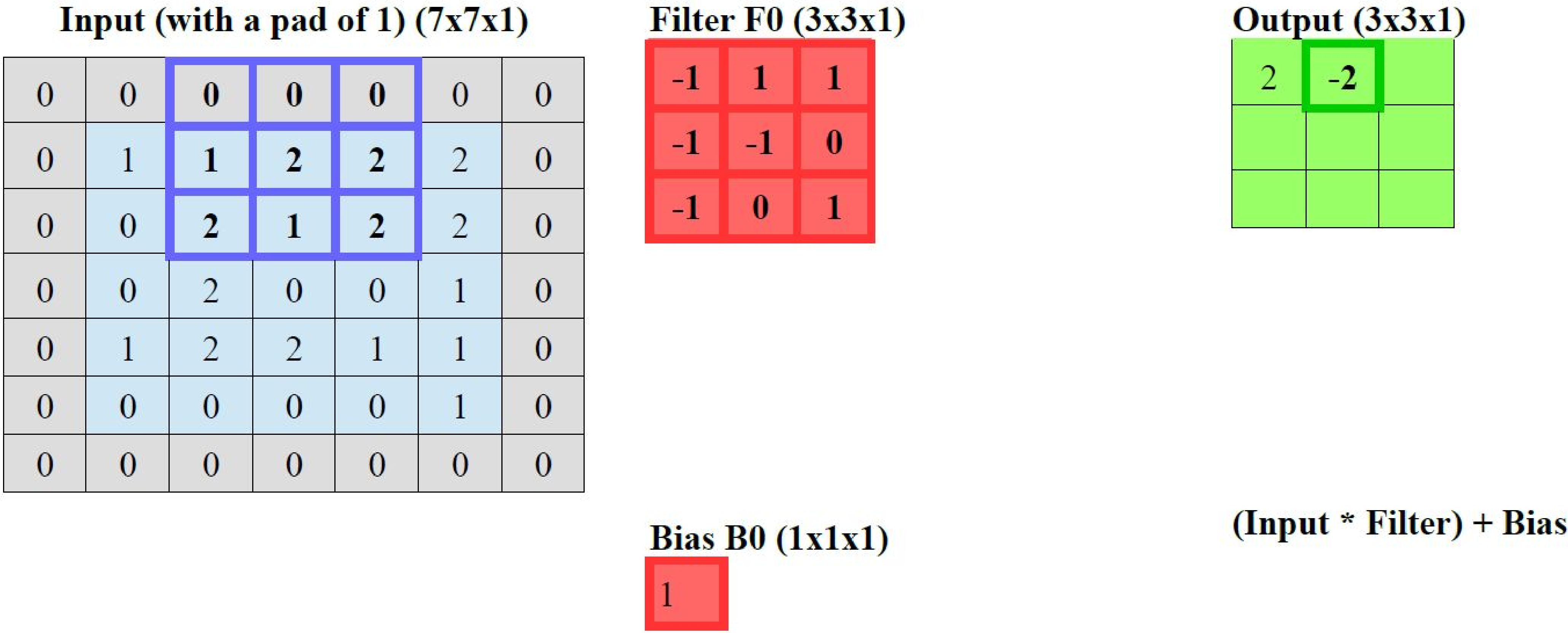
| |
|---|
| 1 |

**Output (3x3x1)**

| | | |
|---|---|---|
| **2** | | |
| | | |
| | | |

**(Input * Filter) + Bias**

# Producing an Activation Map



Input (with a pad of 1) (7x7x1)

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | **0** | **0** | **0** | 0 | 0 |
| 0 | 1 | **1** | **2** | **2** | 2 | 0 |
| 0 | 0 | **2** | **1** | **2** | 2 | 0 |
| 0 | 0 | 2 | 0 | 0 | 1 | 0 |
| 0 | 1 | 2 | 2 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Filter F0 (3x3x1)

| | | |
|---|---|---|
| -1 | 1 | 1 |
| -1 | -1 | 0 |
| -1 | 0 | 1 |

Output (3x3x1)

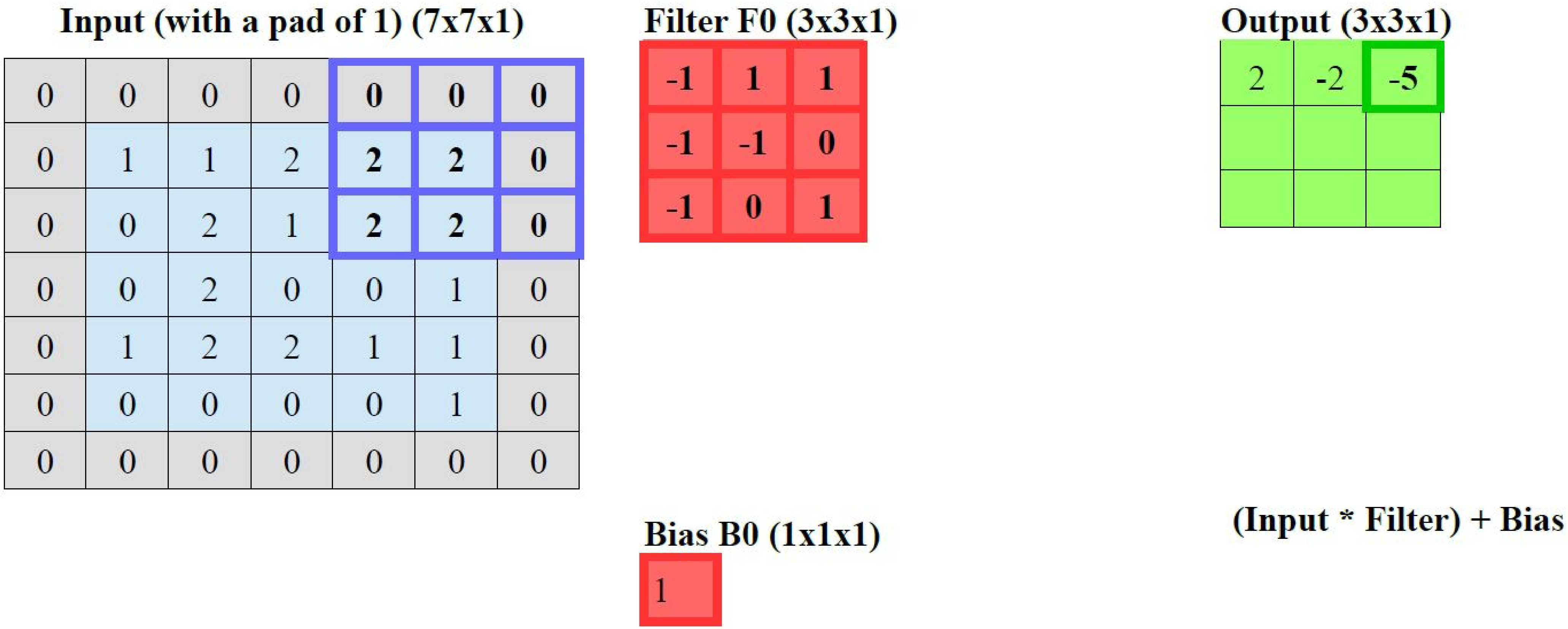
| | | |
|---|---|---|
| 2 | **-2** | |
| | | |
| | | |

Bias B0 (1x1x1)

| |
|---|
| 1 |

(Input * Filter) + Bias

# Producing an Activation Map



**Input (with a pad of 1) (7x7x1)**

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | **0** | **0** | **0** |
| 0 | 1 | 1 | 2 | **2** | **2** | **0** |
| 0 | 0 | 2 | 1 | **2** | **2** | **0** |
| 0 | 0 | 2 | 0 | 0 | 1 | 0 |
| 0 | 1 | 2 | 2 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Filter F0 (3x3x1)**

| | | |
|---|---|---|
| -1 | 1 | 1 |
| -1 | -1 | 0 |
| -1 | 0 | 1 |

**Output (3x3x1)**

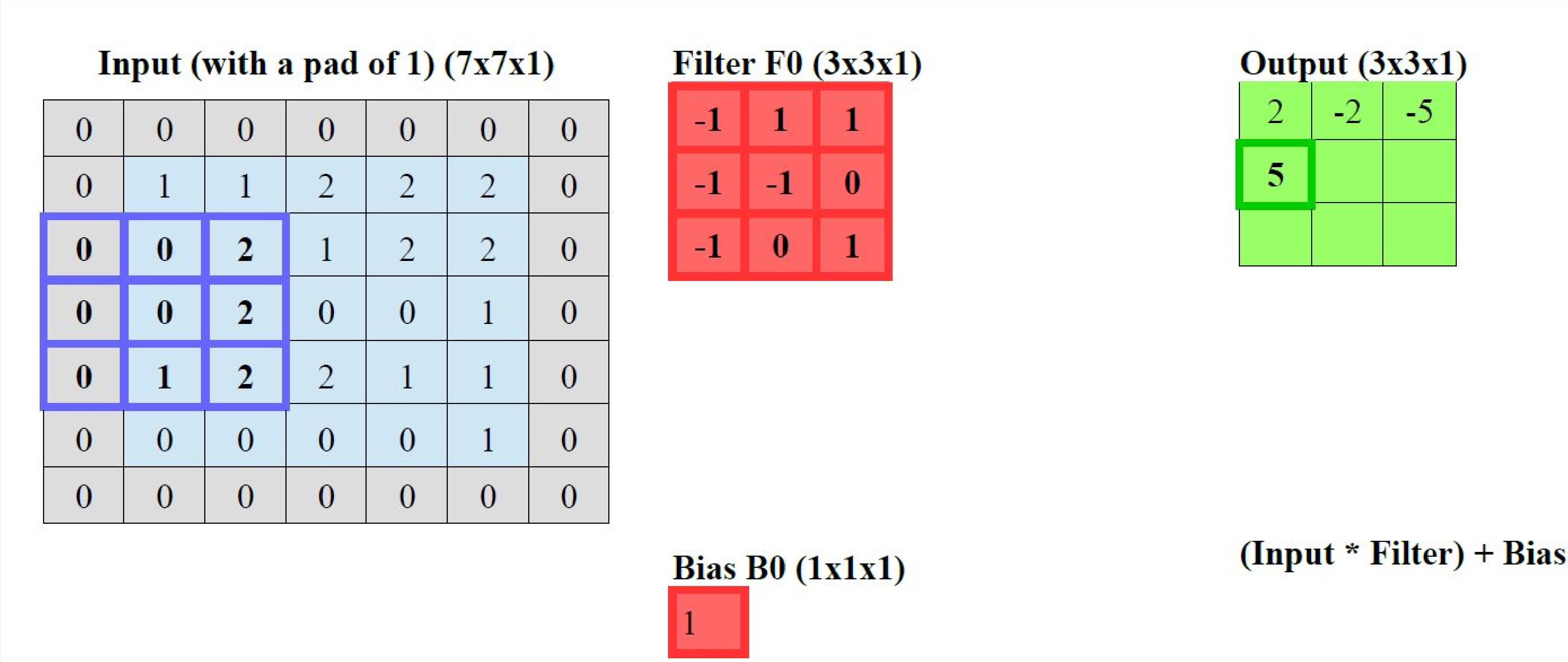
| | | |
|---|---|---|
| 2 | -2 | -5 |
| | | |
| | | |

**Bias B0 (1x1x1)**

| |
|---|
| 1 |

**(Input * Filter) + Bias**

# Producing an Activation Map



Input (with a pad of 1) (7x7x1)

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 2 | 2 | 2 | 0 |
| **0** | **0** | **2** | 1 | 2 | 2 | 0 |
| **0** | **0** | **2** | 0 | 0 | 1 | 0 |
| **0** | **1** | **2** | 2 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Filter F0 (3x3x1)

| | | |
|---|---|---|
| **-1** | **1** | **1** |
| **-1** | **-1** | **0** |
| **-1** | **0** | **1** |

Output (3x3x1)

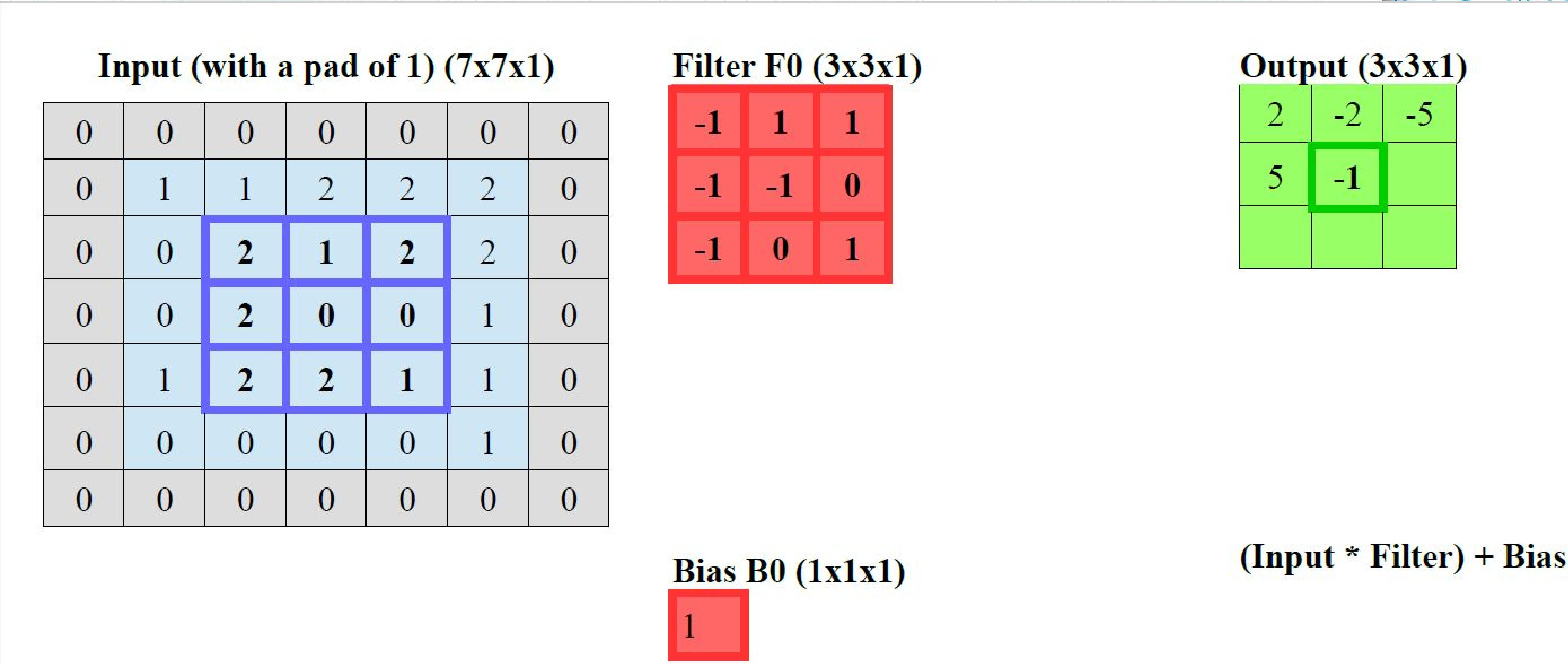
| | | |
|---|---|---|
| 2 | -2 | -5 |
| **5** | | |
| | | |

Bias B0 (1x1x1)

| |
|---|
| 1 |

(Input * Filter) + Bias

# Producing an Activation Map



Input (with a pad of 1) (7x7x1)

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 2 | 2 | 2 | 0 |
| 0 | 0 | **2** | **1** | **2** | 2 | 0 |
| 0 | 0 | **2** | **0** | **0** | 1 | 0 |
| 0 | 1 | **2** | **2** | **1** | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Filter F0 (3x3x1)

| | | |
|---|---|---|
| -1 | 1 | 1 |
| -1 | -1 | 0 |
| -1 | 0 | 1 |

Output (3x3x1)

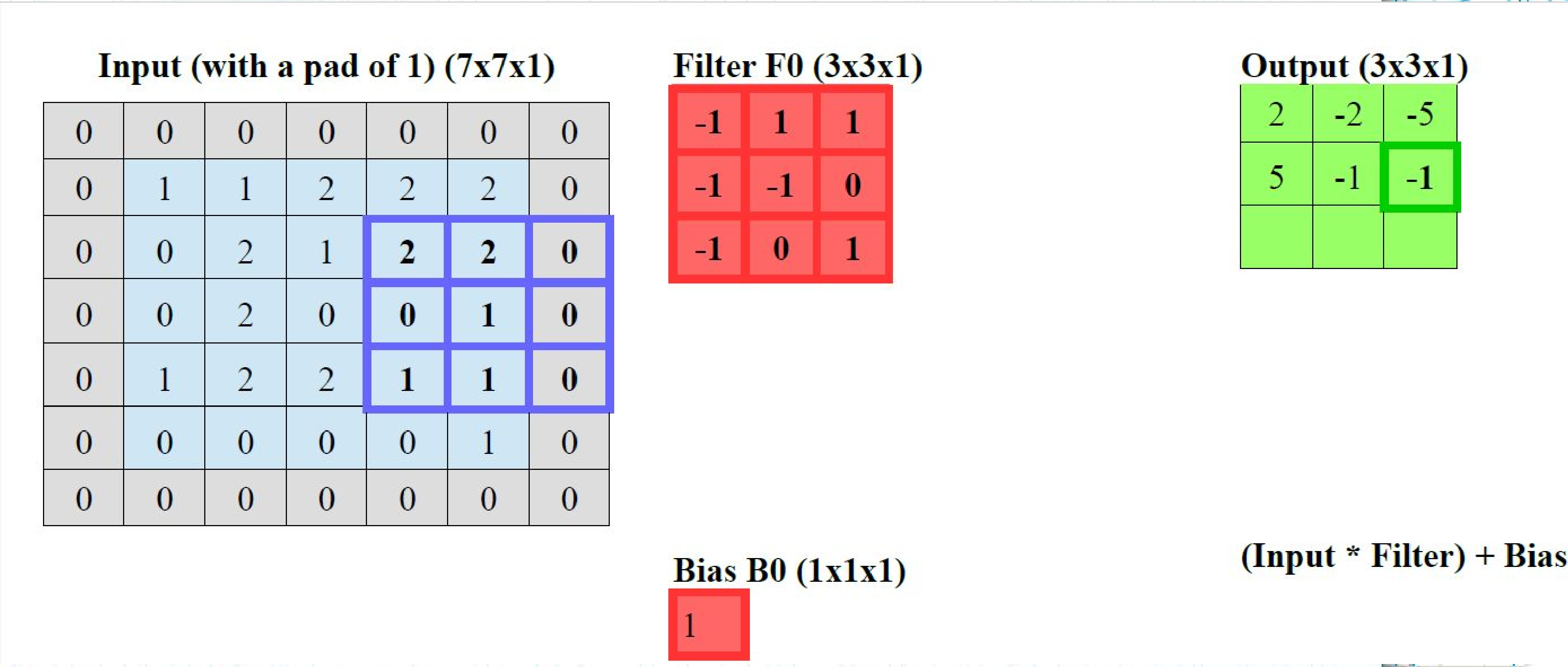
| | | |
|---|---|---|
| 2 | -2 | -5 |
| 5 | -1 | |
| | | |

Bias B0 (1x1x1)

| |
|---|
| 1 |

(Input * Filter) + Bias

# Producing an Activation Map



Input (with a pad of 1) (7x7x1)

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 2 | 2 | 2 | 0 |
| 0 | 0 | 2 | 1 | **2** | **2** | **0** |
| 0 | 0 | 2 | 0 | **0** | **1** | **0** |
| 0 | 1 | 2 | 2 | **1** | **1** | **0** |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Filter F0 (3x3x1)

| | | |
|---|---|---|
| -1 | 1 | 1 |
| -1 | -1 | 0 |
| -1 | 0 | 1 |

Bias B0 (1x1x1)

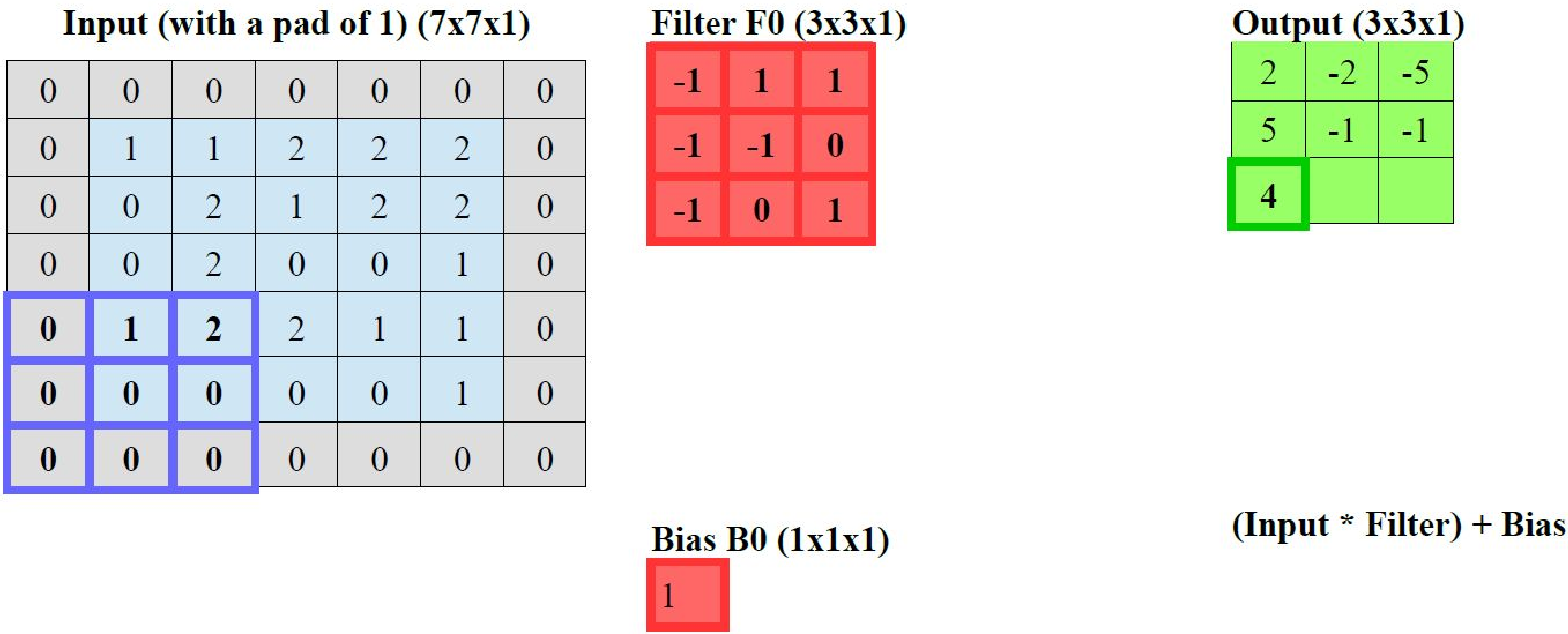
| |
|---|
| 1 |

Output (3x3x1)

| | | |
|---|---|---|
| 2 | -2 | -5 |
| 5 | -1 | -1 |
| | | |

(Input * Filter) + Bias

# Producing an Activation Map



Input (with a pad of 1) (7x7x1)

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 2 | 2 | 2 | 0 |
| 0 | 0 | 2 | 1 | 2 | 2 | 0 |
| 0 | 0 | 2 | 0 | 0 | 1 | 0 |
| **0** | **1** | **2** | 2 | 1 | 1 | 0 |
| **0** | **0** | **0** | 0 | 0 | 1 | 0 |
| **0** | **0** | **0** | 0 | 0 | 0 | 0 |

Filter F0 (3x3x1)

| | | |
|---|---|---|
| **-1** | **1** | **1** |
| **-1** | **-1** | **0** |
| **-1** | **0** | **1** |

Bias B0 (1x1x1)

| |
|---|
| 1 |

Output (3x3x1)

| | | |
|---|---|---|
| 2 | -2 | -5 |
| 5 | -1 | -1 |
| 4 | | |

(Input * Filter) + Bias

# Producing an Activation Map

**Input (with a pad of 1) (7x7x1)**

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 2 | 2 | 2 | 0 |
| 0 | 0 | 2 | 1 | 2 | 2 | 0 |
| 0 | 0 | 2 | 0 | 0 | 1 | 0 |
| 0 | 1 | **2** | **2** | **1** | 1 | 0 |
| 0 | 0 | **0** | **0** | **0** | 1 | 0 |
| 0 | 0 | **0** | **0** | **0** | 0 | 0 |

**Filter F0 (3x3x1)**

| | | |
|---|---|---|
| **-1** | **1** | **1** |
| **-1** | **-1** | **0** |
| **-1** | **0** | **1** |

**Bias B0 (1x1x1)**

| |
|---|
| 1 |

**Output (3x3x1)**

| | | |
|---|---|---|
| 2 | -2 | -5 |
| 5 | -1 | -1 |
| 4 | **2** | |

**(Input * Filter) + Bias**

# Producing an Activation Map

**Input (with a pad of 1) (7x7x1)**

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 2 | 2 | 2 | 0 |
| 0 | 0 | 2 | 1 | 2 | 2 | 0 |
| 0 | 0 | 2 | 0 | 0 | 1 | 0 |
| 0 | 1 | 2 | 2 | **1** | **1** | **0** |
| 0 | 0 | 0 | 0 | **0** | **1** | **0** |
| 0 | 0 | 0 | 0 | **0** | **0** | **0** |

**Filter F0 (3x3x1)**

| | | |
|---|---|---|
| **-1** | **1** | **1** |
| **-1** | **-1** | **0** |
| **-1** | **0** | **1** |

**Bias B0 (1x1x1)**

| |
|---|
| 1 |

**Output (3x3x1)**

| | | |
|---|---|---|
| 2 | -2 | -5 |
| 5 | -1 | -1 |
| 4 | 2 | **0** |

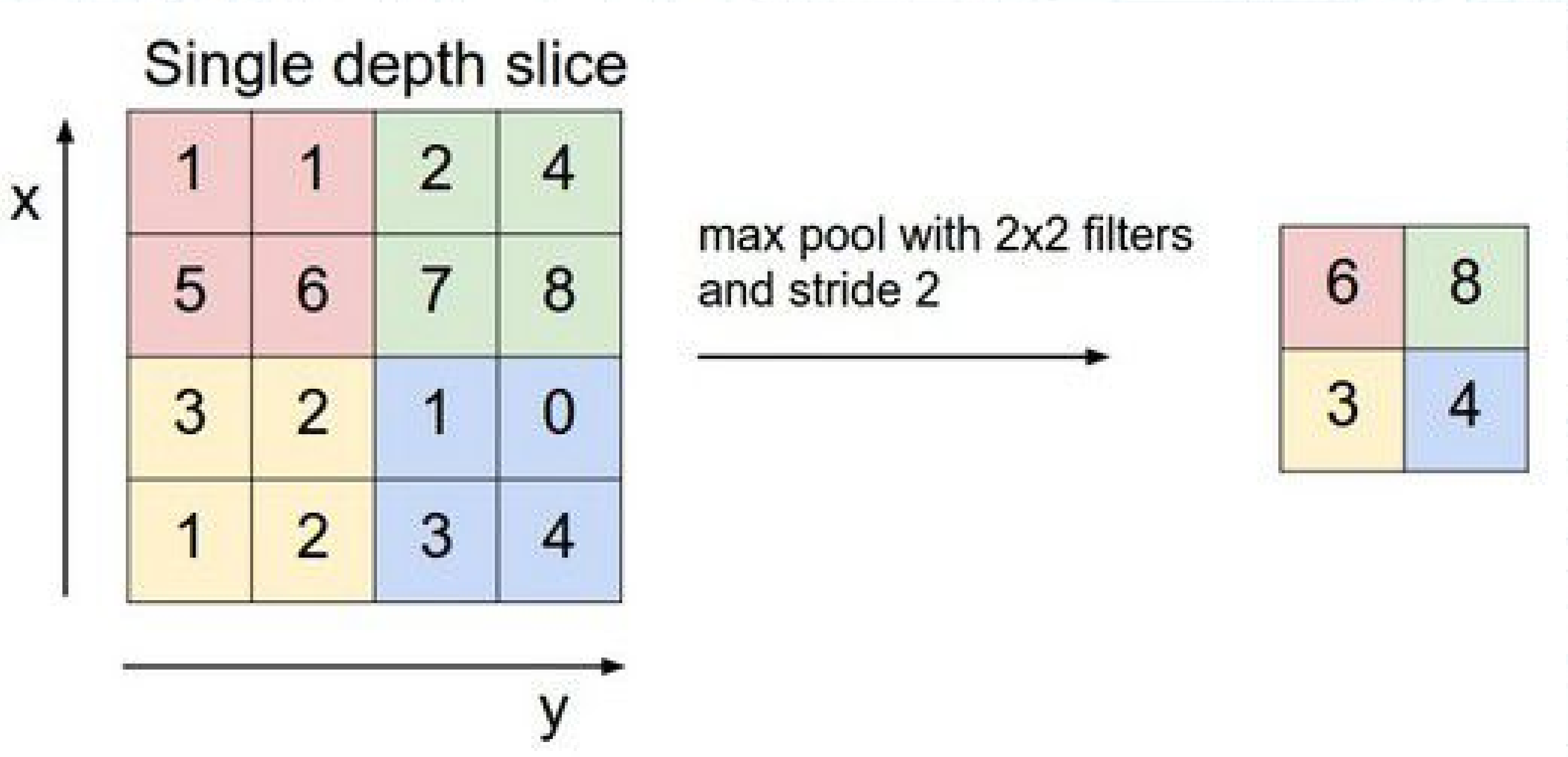
**(Input * Filter) + Bias**

# Convolutional Network Parameters

- Parameters and their effects
  - Kernel Size
  - Stride length
  - Number of Filters
  - Depth of classifying network
- Optional layers and features
  - Downsampling
  - Dropout technique

# The Downsampling Layer

- Decrease computational complexity



Single depth slice

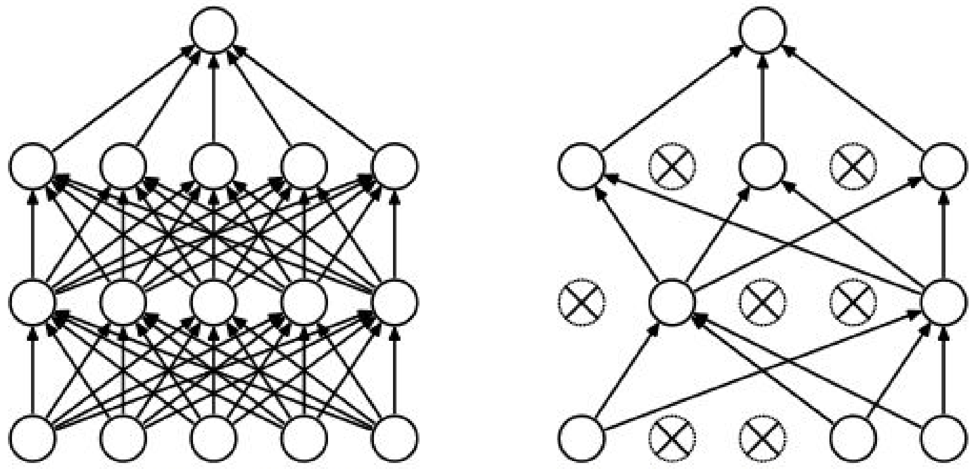
| | | | |
|---|---|---|---|
| 1 | 1 | 2 | 4 |
| 5 | 6 | 7 | 8 |
| 3 | 2 | 1 | 0 |
| 1 | 2 | 3 | 4 |

max pool with 2x2 filters and stride 2

| | |
|---|---|
| 6 | 8 |
| 3 | 4 |

# Dropout Technique

- Temporarily disable neurons
- Prevents overfitting
- Only used in dense layers

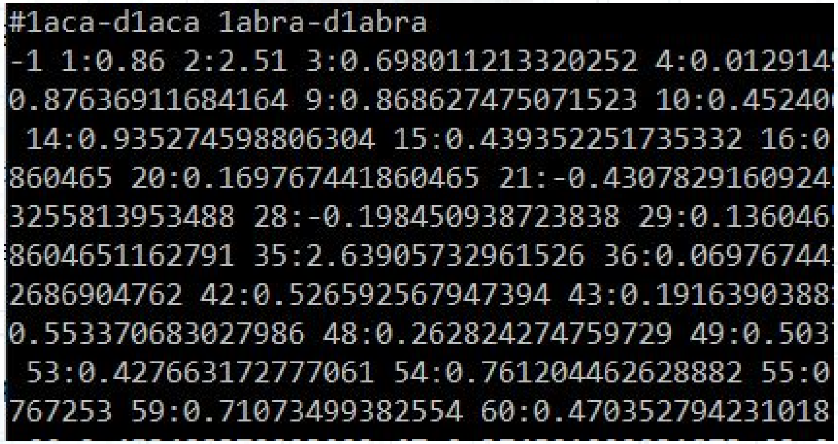


(a) Standard Neural Net

(b) After applying dropout.

# Dataset

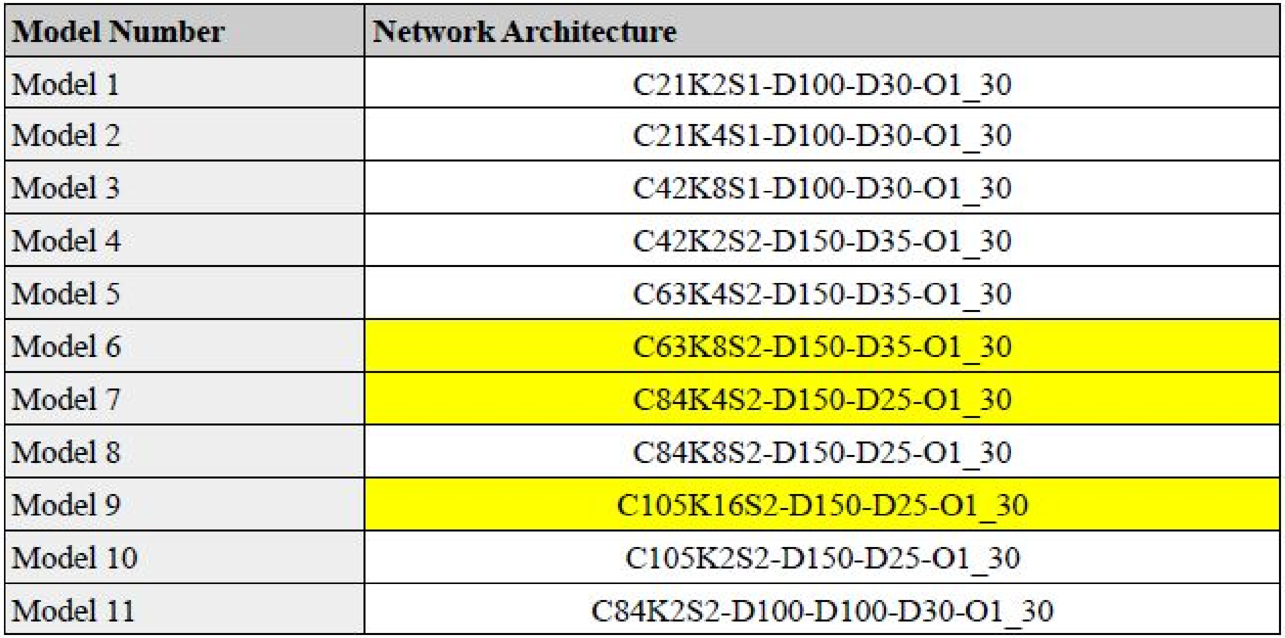
- Derived from the SCOP database
- Proteins
  - Family (555)
  - Superfamily (434)
  - Fold (321)
- Sequence, profile, family alignment and structural information
- 84 data points



```
#1aca-d1aca 1abra-d1abra
-1 1:0.86 2:2.51 3:0.698011213320252 4:0.012914
0.87636911684164 9:0.868627475071523 10:0.45240
 14:0.935274598806304 15:0.439352251735332 16:0
860465 20:0.169767441860465 21:-0.4307829160924
3255813953488 28:-0.198450938723838 29:0.136046
8604651162791 35:2.63905732961526 36:0.06976744
2686904762 42:0.526592567947394 43:0.1916390388
0.553370683027986 48:0.262824274759729 49:0.503
 53:0.427663172777061 54:0.761204462628882 55:0
767253 59:0.71073499382554 60:0.470352794231018
```

# Models

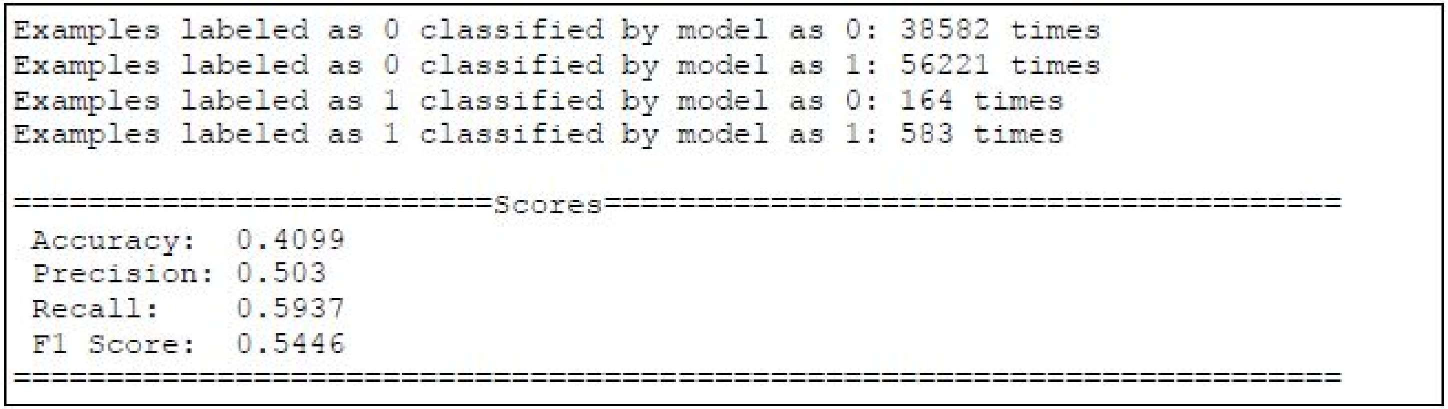
- Generated 11 simi-random networks
- Varying kernels, strides, classifying networks



| Model Number | Network Architecture |
|---|---|
| Model 1 | C21K2S1-D100-D30-O1_30 |
| Model 2 | C21K4S1-D100-D30-O1_30 |
| Model 3 | C42K8S1-D100-D30-O1_30 |
| Model 4 | C42K2S2-D150-D35-O1_30 |
| Model 5 | C63K4S2-D150-D35-O1_30 |
| Model 6 | C63K8S2-D150-D35-O1_30 |
| Model 7 | C84K4S2-D150-D25-O1_30 |
| Model 8 | C84K8S2-D150-D25-O1_30 |
| Model 9 | C105K16S2-D150-D25-O1_30 |
| Model 10 | C105K2S2-D150-D25-O1_30 |
| Model 11 | C84K2S2-D100-D100-D30-O1_30 |

# Model Selection

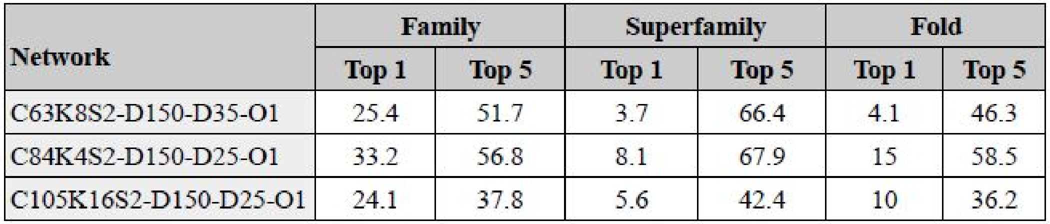
- Initial testing on a randomized test set
- Chose the top 3 networks to train and test



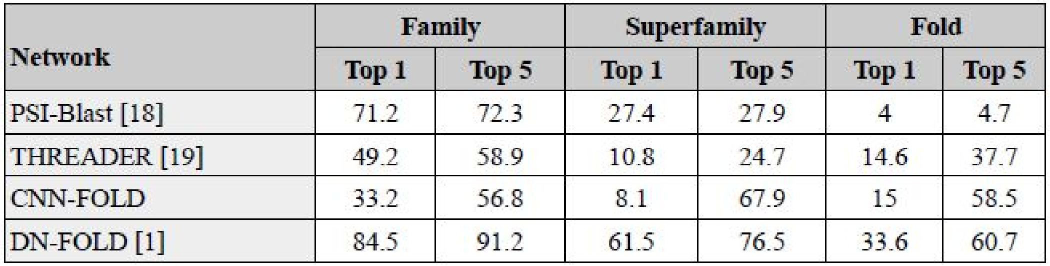
```
Examples labeled as 0 classified by model as 0: 38582 times
Examples labeled as 0 classified by model as 1: 56221 times
Examples labeled as 1 classified by model as 0: 164 times
Examples labeled as 1 classified by model as 1: 583 times

==========================Scores========================================
 Accuracy:  0.4099
 Precision: 0.503
 Recall:    0.5937
 F1 Score:  0.5446
========================================================================
```

# Results

- Overall CNNs did not outperform DN-Fold
- Provided comparable results to past methods
- Data format
- Label balance



| Network | Family | | Superfamily | | Fold | |
|---|---|---|---|---|---|---|
| | Top 1 | Top 5 | Top 1 | Top 5 | Top 1 | Top 5 |
| C63K8S2-D150-D35-O1 | 25.4 | 51.7 | 3.7 | 66.4 | 4.1 | 46.3 |
| C84K4S2-D150-D25-O1 | 33.2 | 56.8 | 8.1 | 67.9 | 15 | 58.5 |
| C105K16S2-D150-D25-O1 | 24.1 | 37.8 | 5.6 | 42.4 | 10 | 36.2 |



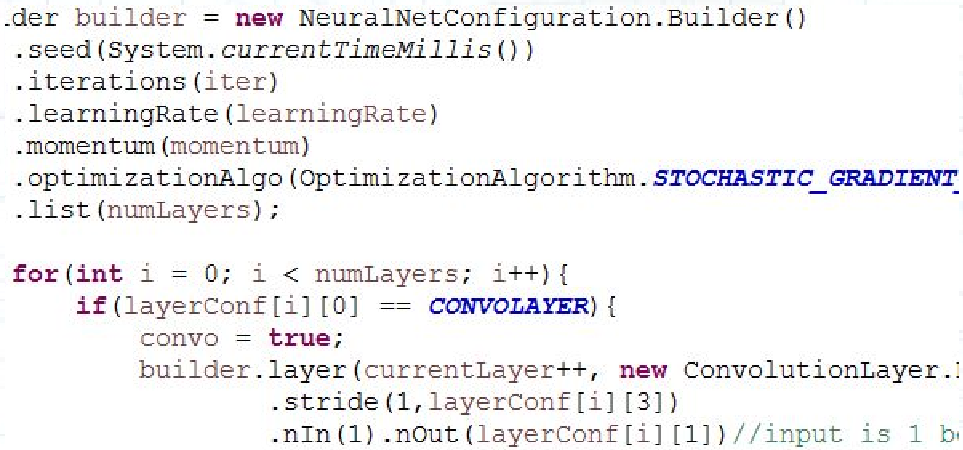
| Network | Family | | Superfamily | | Fold | |
|---|---|---|---|---|---|---|
| | Top 1 | Top 5 | Top 1 | Top 5 | Top 1 | Top 5 |
| PSI-Blast [18] | 71.2 | 72.3 | 27.4 | 27.9 | 4 | 4.7 |
| THREADER [19] | 49.2 | 58.9 | 10.8 | 24.7 | 14.6 | 37.7 |
| CNN-FOLD | 33.2 | 56.8 | 8.1 | 67.9 | 15 | 58.5 |
| DN-FOLD [1] | 84.5 | 91.2 | 61.5 | 76.5 | 33.6 | 60.7 |

# Conclusions

- Tasked with CNNs applied to DN-Fold Dataset
- Lacked spacial properties
- More filters not always good

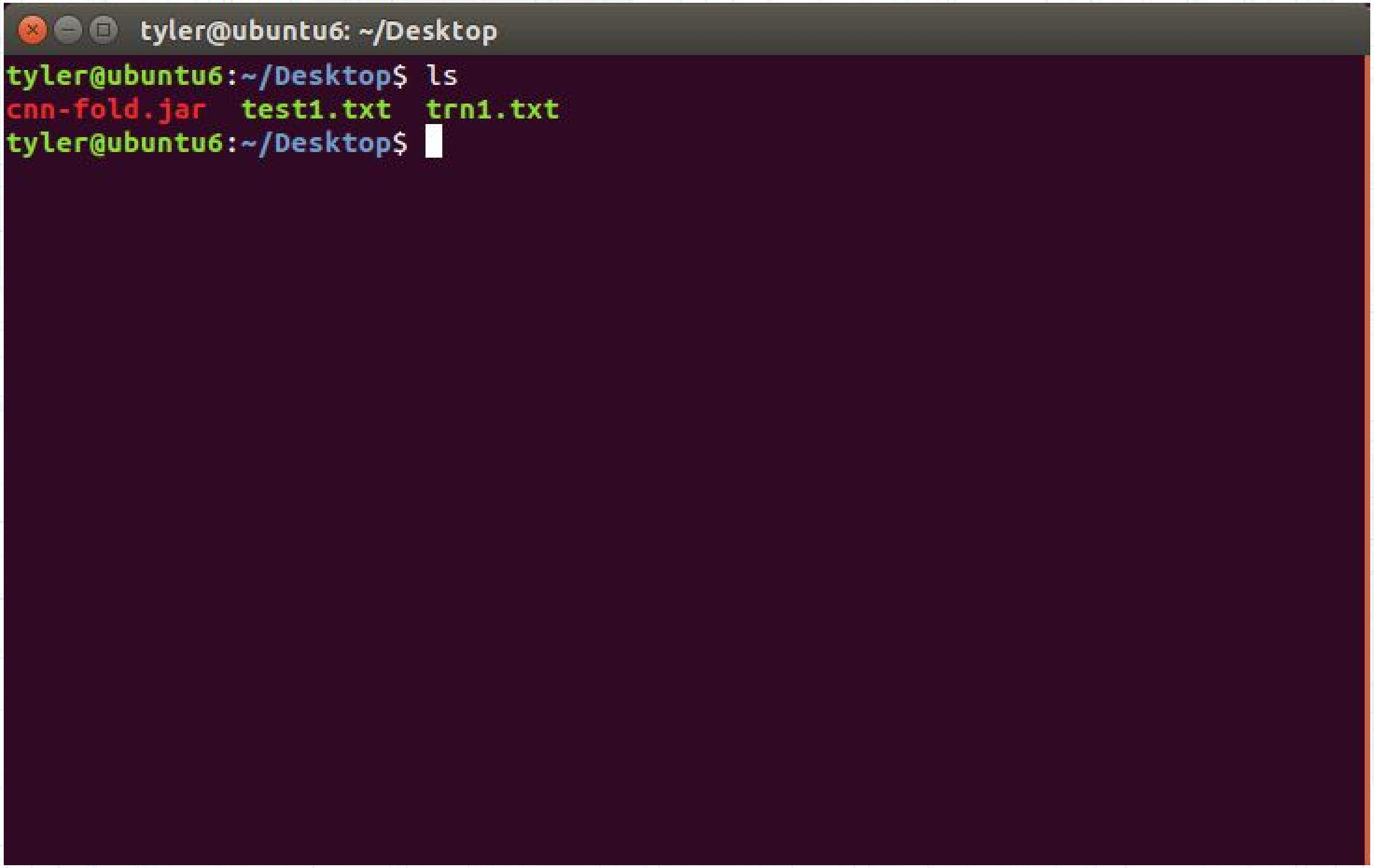
# CNN-Fold

- Program written to obtain results
- Specify CNN-Fold, DN-Fold, Command, or Json network architectures
- Train and Test modes
- Saves trained networks
- Download



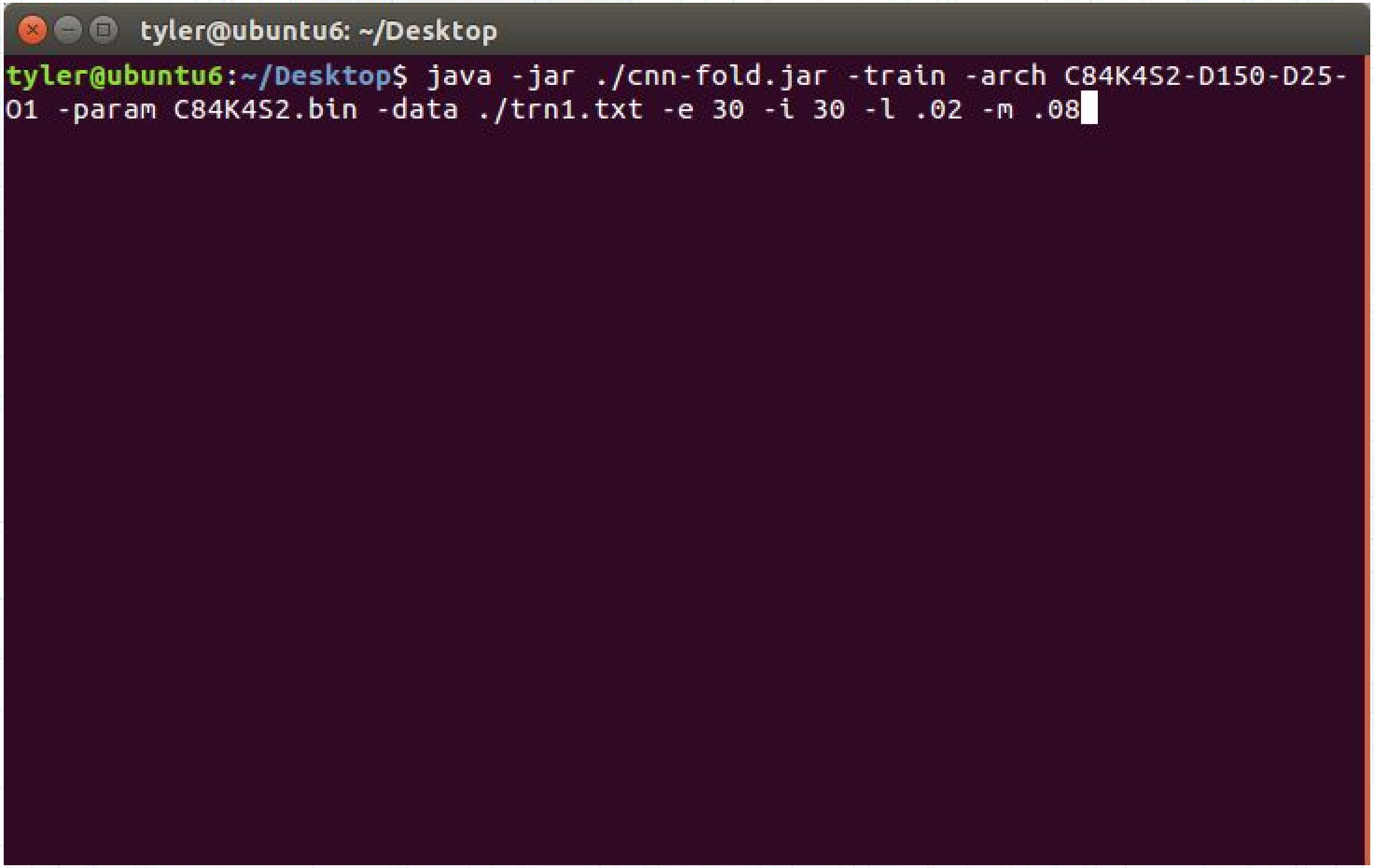
```
der builder = new NeuralNetConfiguration.Builder()
 .seed(System.currentTimeMillis())
 .iterations(iter)
 .learningRate(learningRate)
 .momentum(momentum)
 .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT
 .list(numLayers);

 for(int i = 0; i < numLayers; i++){
     if(layerConf[i][0] == CONVOLAYER){
         convo = true;
         builder.layer(currentLayer++, new ConvolutionLayer.
                 .stride(1,layerConf[i][3])
                 .nIn(1).nOut(layerConf[i][1])//input is 1 b
```

# CNN-Fold Demo



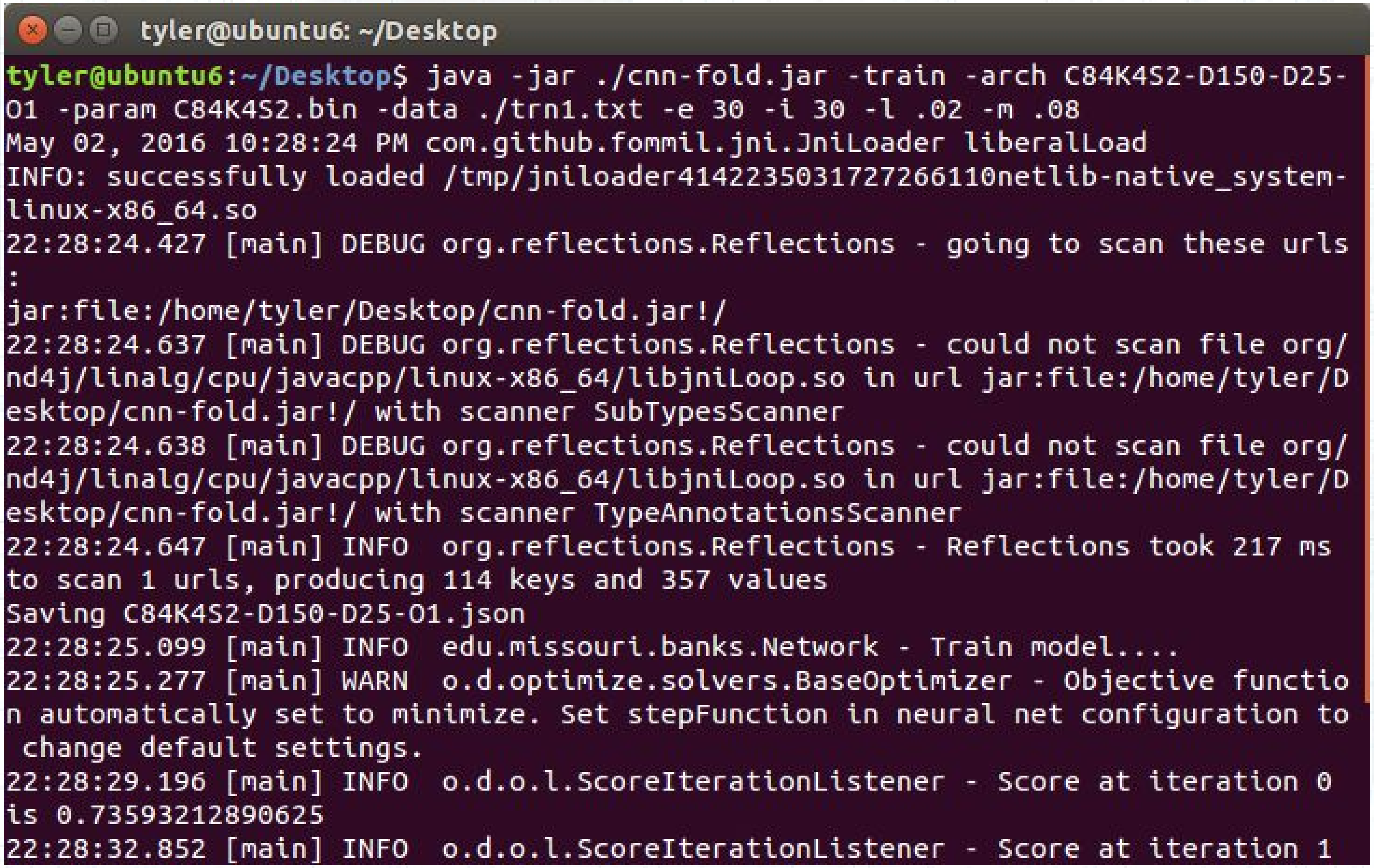
```
tyler@ubuntu6:~/Desktop$ ls
cnn-fold.jar  test1.txt  trn1.txt
tyler@ubuntu6:~/Desktop$ 
```

# CNN-Fold Demo



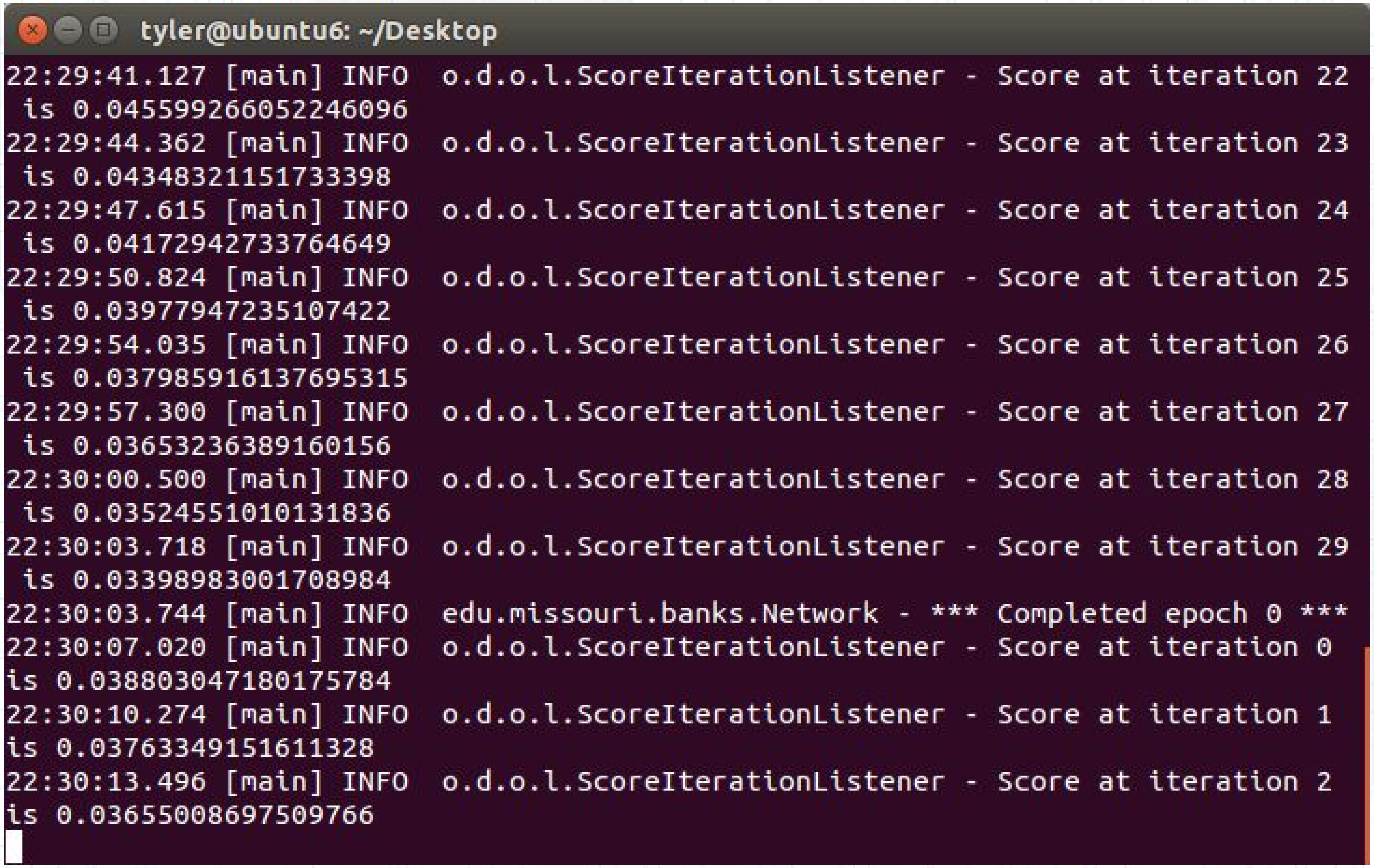
```
tyler@ubuntu6:~/Desktop$ java -jar ./cnn-fold.jar -train -arch C84K4S2-D150-D25-
O1 -param C84K4S2.bin -data ./trn1.txt -e 30 -i 30 -l .02 -m .08
```

# CNN-Fold Demo



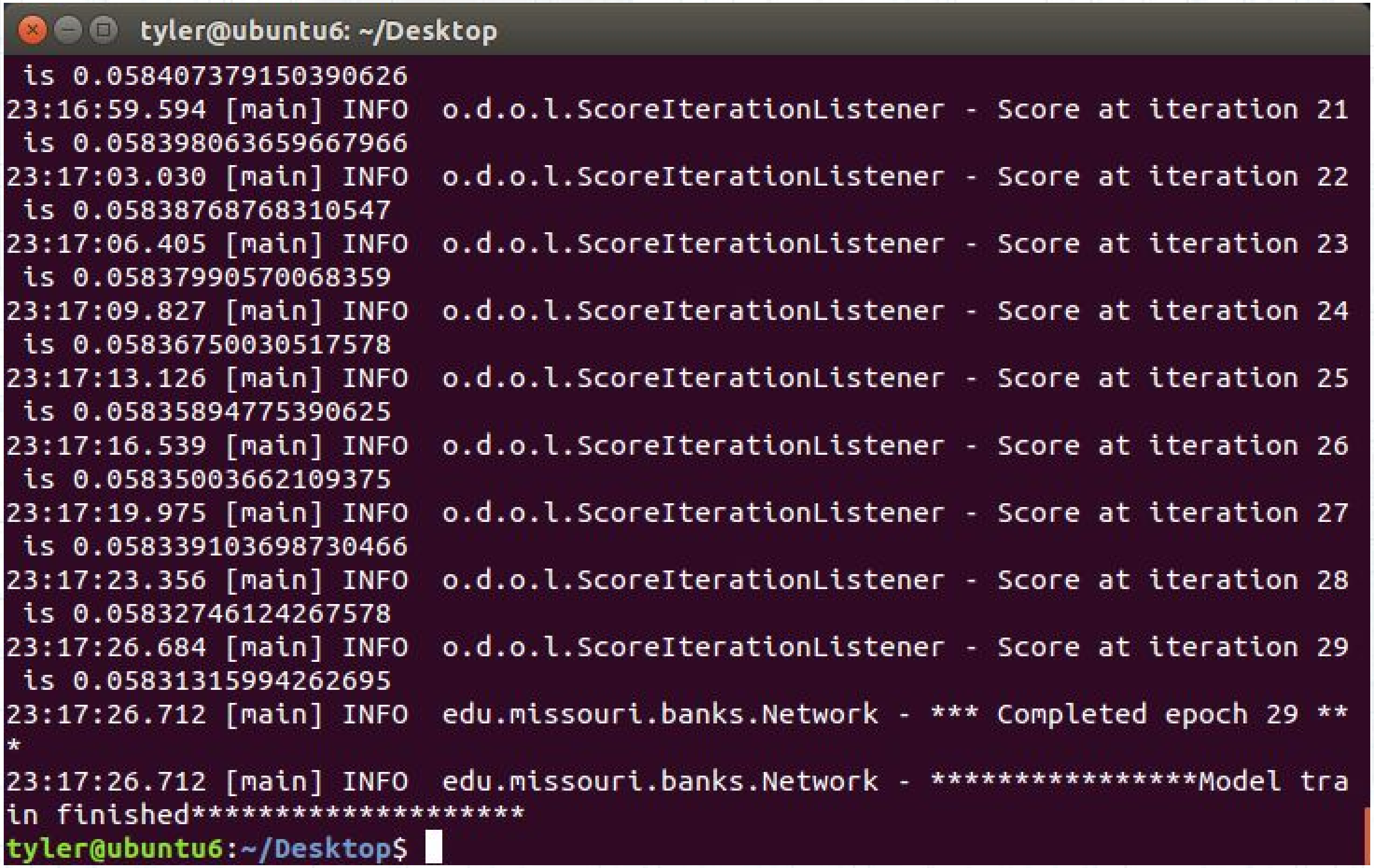
```
tyler@ubuntu6:~/Desktop$ java -jar ./cnn-fold.jar -train -arch C84K4S2-D150-D25-
01 -param C84K4S2.bin -data ./trn1.txt -e 30 -i 30 -l .02 -m .08
May 02, 2016 10:28:24 PM com.github.fommil.jni.JniLoader liberalLoad
INFO: successfully loaded /tmp/jniloader4142235031727266110netlib-native_system-
linux-x86_64.so
22:28:24.427 [main] DEBUG org.reflections.Reflections - going to scan these urls
:
jar:file:/home/tyler/Desktop/cnn-fold.jar!/
22:28:24.637 [main] DEBUG org.reflections.Reflections - could not scan file org/
nd4j/linalg/cpu/javacpp/linux-x86_64/libjniLoop.so in url jar:file:/home/tyler/D
esktop/cnn-fold.jar!/ with scanner SubTypesScanner
22:28:24.638 [main] DEBUG org.reflections.Reflections - could not scan file org/
nd4j/linalg/cpu/javacpp/linux-x86_64/libjniLoop.so in url jar:file:/home/tyler/D
esktop/cnn-fold.jar!/ with scanner TypeAnnotationsScanner
22:28:24.647 [main] INFO  org.reflections.Reflections - Reflections took 217 ms
to scan 1 urls, producing 114 keys and 357 values
Saving C84K4S2-D150-D25-01.json
22:28:25.099 [main] INFO  edu.missouri.banks.Network - Train model....
22:28:25.277 [main] WARN  o.d.optimize.solvers.BaseOptimizer - Objective functio
n automatically set to minimize. Set stepFunction in neural net configuration to
 change default settings.
22:28:29.196 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 0
is 0.73593212890625
22:28:32.852 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 1
```

# CNN-Fold Demo



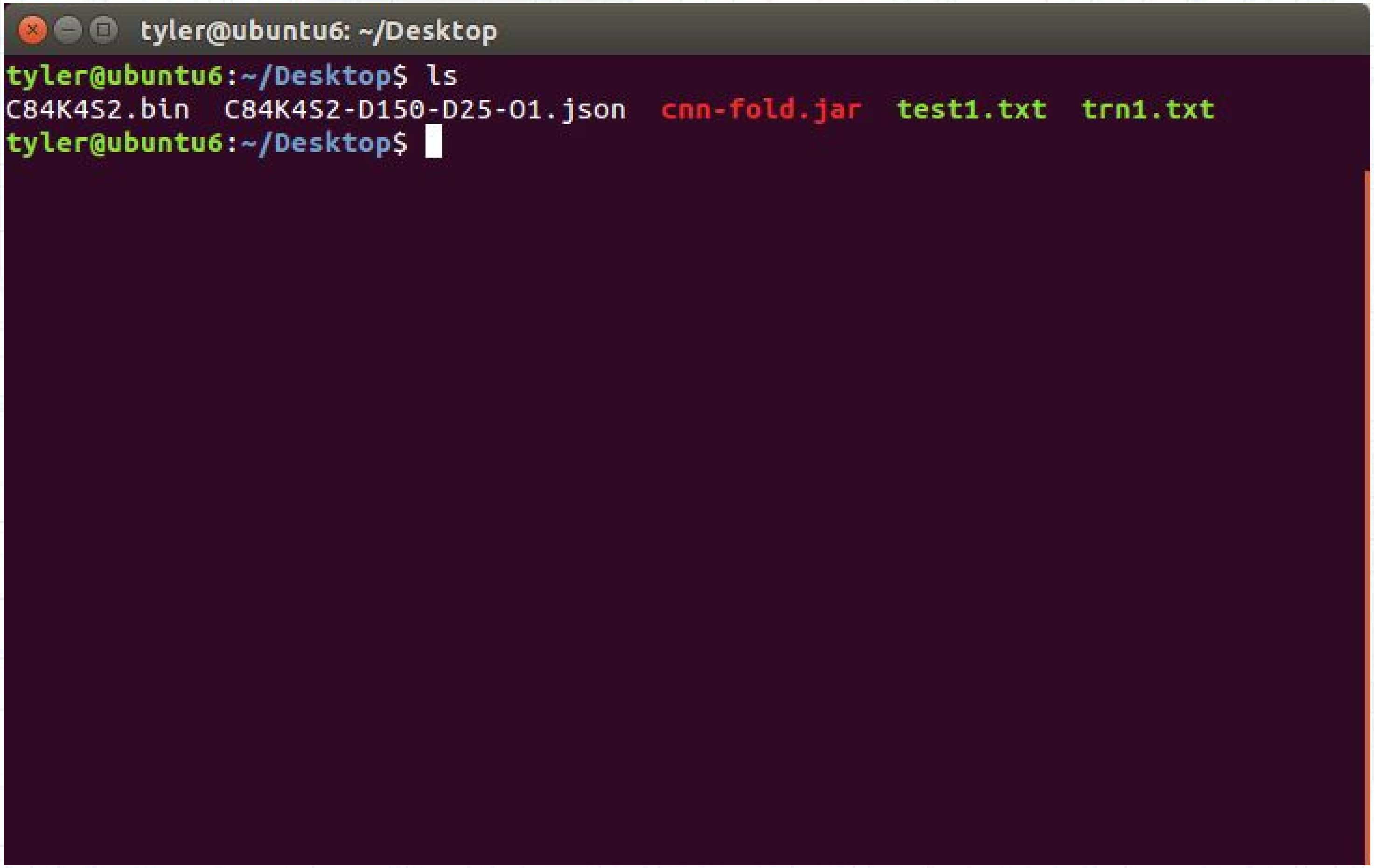
```
22:29:41.127 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 22
 is 0.045599266052246096
22:29:44.362 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 23
 is 0.04348321151733398
22:29:47.615 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 24
 is 0.04172942733764649
22:29:50.824 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 25
 is 0.03977947235107422
22:29:54.035 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 26
 is 0.037985916137695315
22:29:57.300 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 27
 is 0.03653236389160156
22:30:00.500 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 28
 is 0.03524551010131836
22:30:03.718 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 29
 is 0.03398983001708984
22:30:03.744 [main] INFO  edu.missouri.banks.Network - *** Completed epoch 0 ***
22:30:07.020 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 0
is 0.038803047180175784
22:30:10.274 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 1
is 0.03763349151611328
22:30:13.496 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 2
is 0.03655008697509766
```

# CNN-Fold Demo



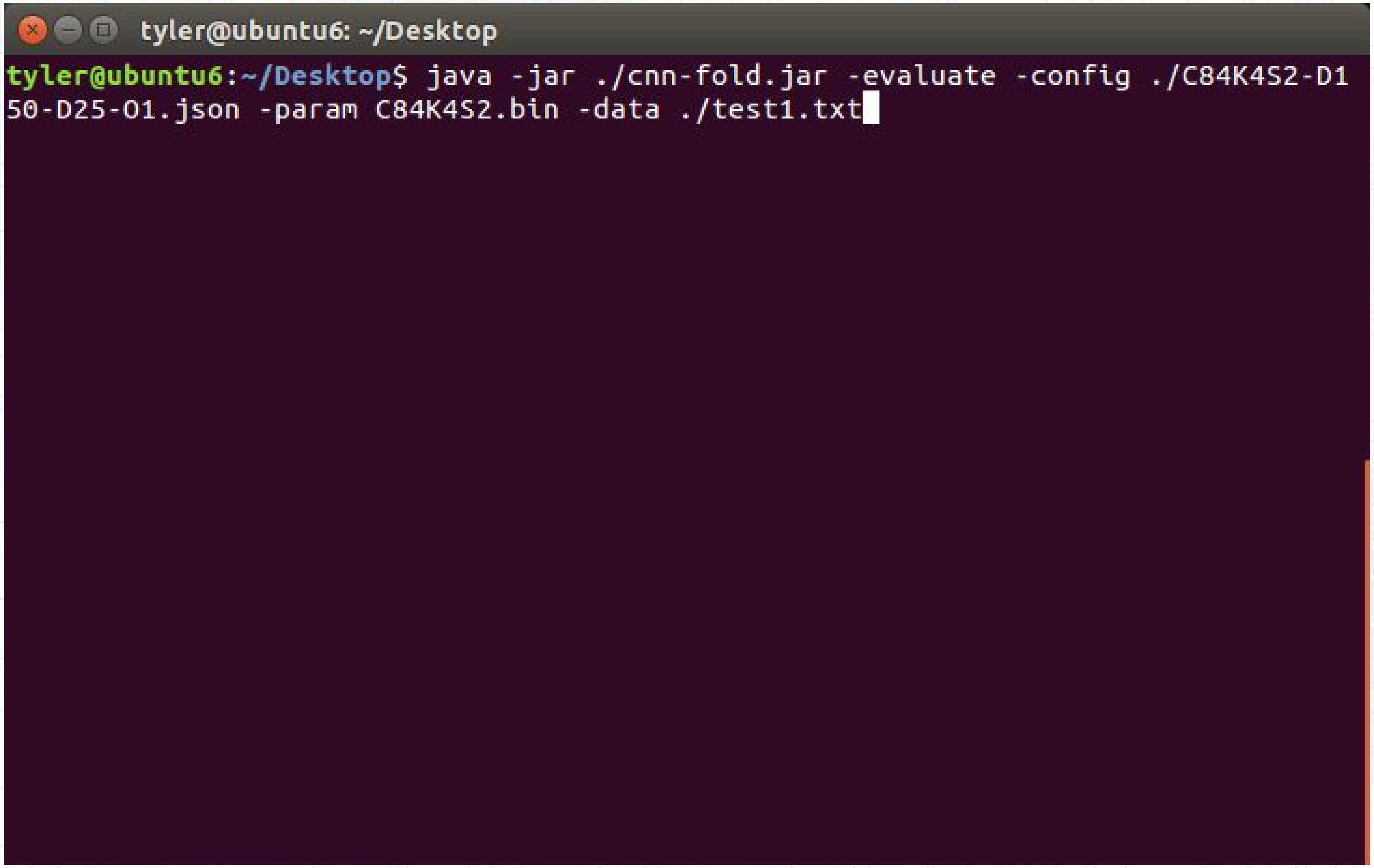
```
 is 0.05840737915039062б
23:16:59.594 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 21
 is 0.05839806365966796б
23:17:03.030 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 22
 is 0.0583876876831054 7
23:17:06.405 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 23
 is 0.05837990570068359
23:17:09.827 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 24
 is 0.0583675003051757 8
23:17:13.126 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 25
 is 0.058358947753906 25
23:17:16.539 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 26
 is 0.05835003662109375
23:17:19.975 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 27
 is 0.05833910369873046б
23:17:23.356 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 28
 is 0.0583274612426757 8
23:17:26.684 [main] INFO  o.d.o.l.ScoreIterationListener - Score at iteration 29
 is 0.05831315994262695
23:17:26.712 [main] INFO  edu.missouri.banks.Network - *** Completed epoch 29 **
*
23:17:26.712 [main] INFO  edu.missouri.banks.Network - ****************Model tra
in finished********************
tyler@ubuntu6:~/Desktop$ 
```

# CNN-Fold Demo



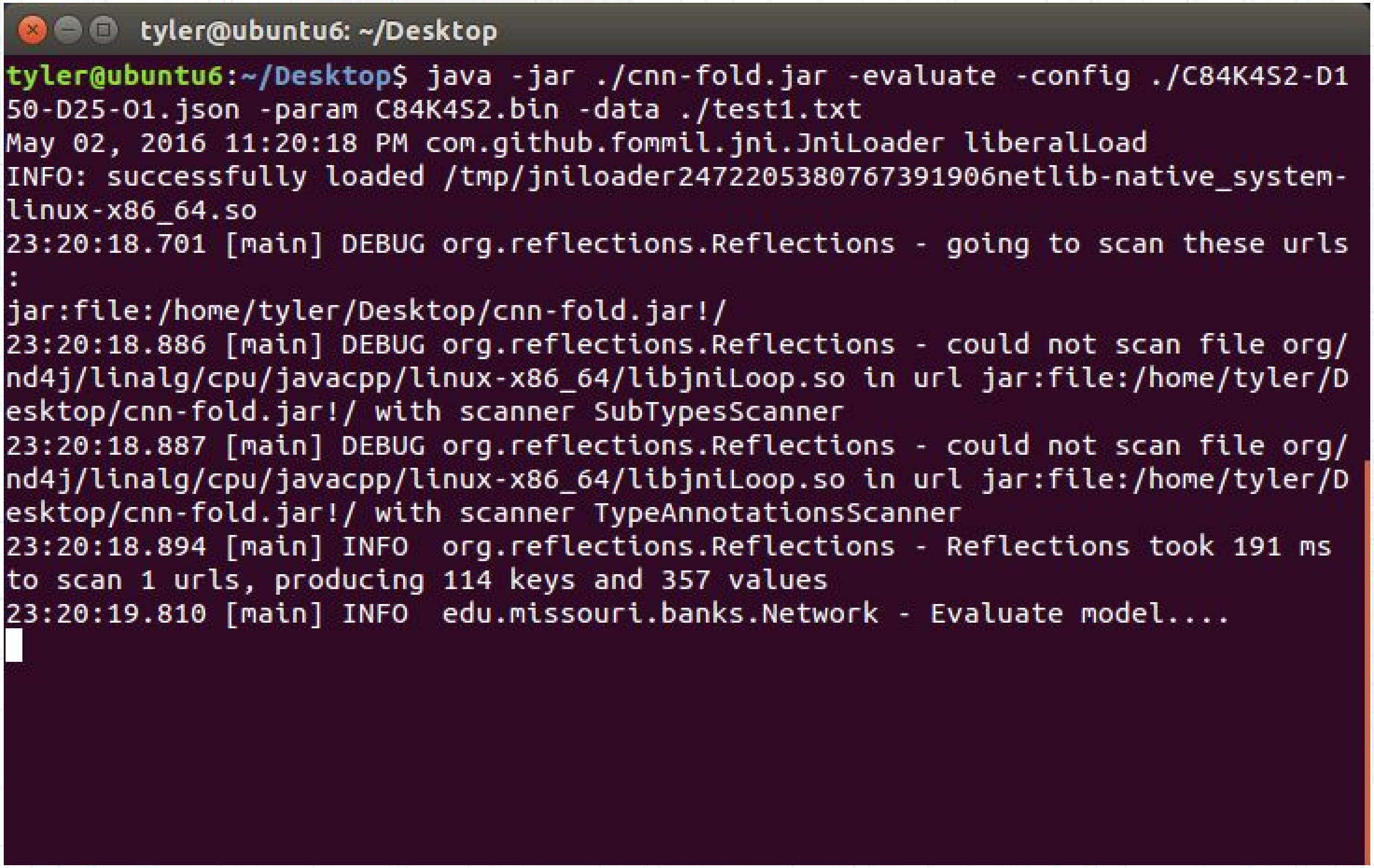
```
tyler@ubuntu6:~/Desktop$ ls
C84K4S2.bin  C84K4S2-D150-D25-O1.json  cnn-fold.jar  test1.txt  trn1.txt
tyler@ubuntu6:~/Desktop$ 
```

# CNN-Fold Demo



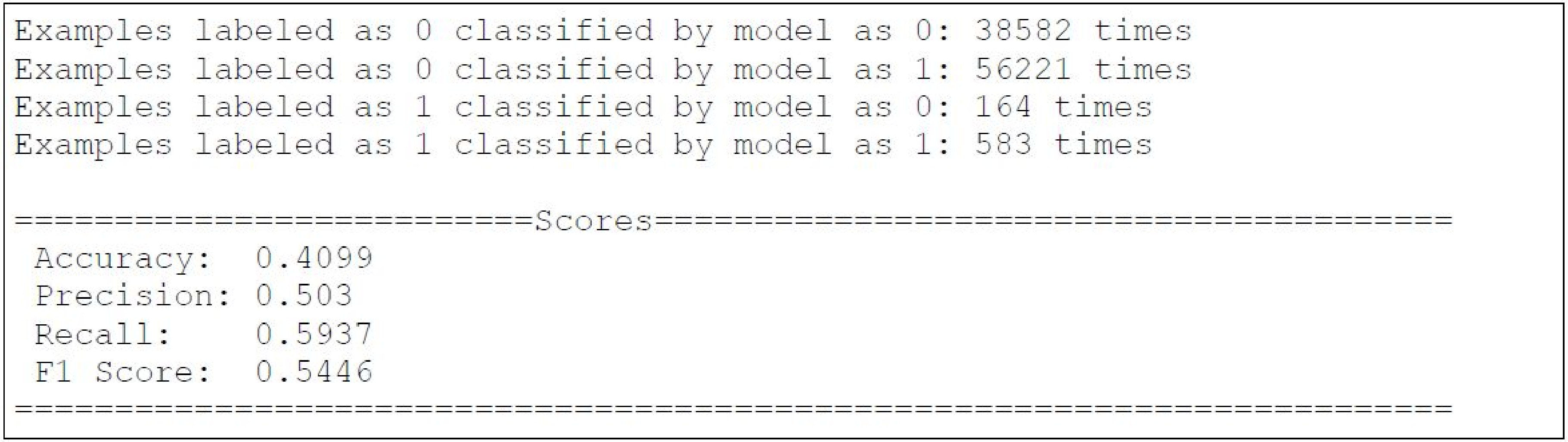
```
tyler@ubuntu6:~/Desktop$ java -jar ./cnn-fold.jar -evaluate -config ./C84K4S2-D1
50-D25-O1.json -param C84K4S2.bin -data ./test1.txt
```

# CNN-Fold Demo



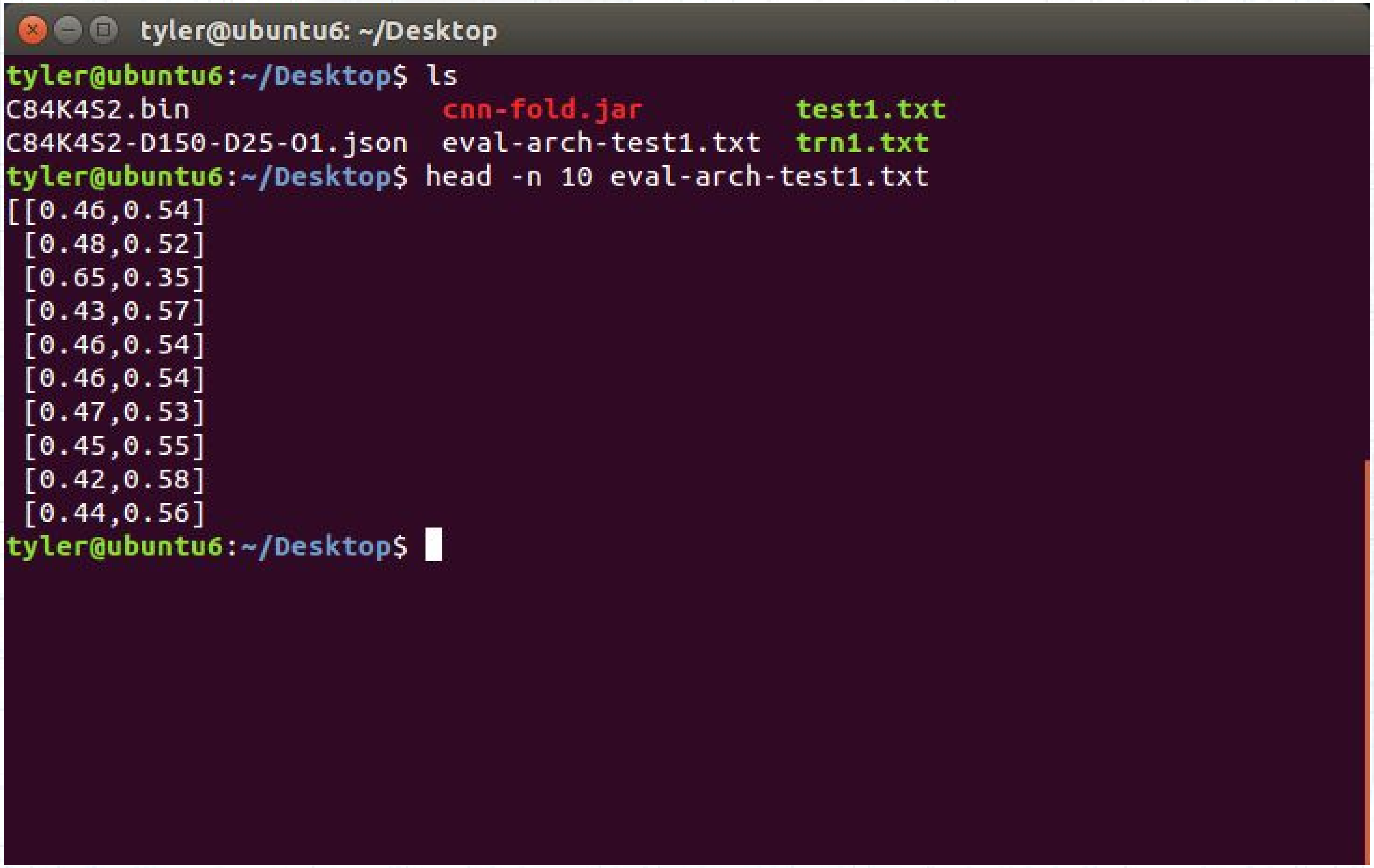
```
tyler@ubuntu6:~/Desktop$ java -jar ./cnn-fold.jar -evaluate -config ./C84K4S2-D1
50-D25-O1.json -param C84K4S2.bin -data ./test1.txt
May 02, 2016 11:20:18 PM com.github.fommil.jni.JniLoader liberalLoad
INFO: successfully loaded /tmp/jniloader2472205380767391906netlib-native_system-
linux-x86_64.so
23:20:18.701 [main] DEBUG org.reflections.Reflections - going to scan these urls
:
jar:file:/home/tyler/Desktop/cnn-fold.jar!/
23:20:18.886 [main] DEBUG org.reflections.Reflections - could not scan file org/
nd4j/linalg/cpu/javacpp/linux-x86_64/libjniLoop.so in url jar:file:/home/tyler/D
esktop/cnn-fold.jar!/ with scanner SubTypesScanner
23:20:18.887 [main] DEBUG org.reflections.Reflections - could not scan file org/
nd4j/linalg/cpu/javacpp/linux-x86_64/libjniLoop.so in url jar:file:/home/tyler/D
esktop/cnn-fold.jar!/ with scanner TypeAnnotationsScanner
23:20:18.894 [main] INFO  org.reflections.Reflections - Reflections took 191 ms
to scan 1 urls, producing 114 keys and 357 values
23:20:19.810 [main] INFO  edu.missouri.banks.Network - Evaluate model....
```

# CNN-Fold Demo



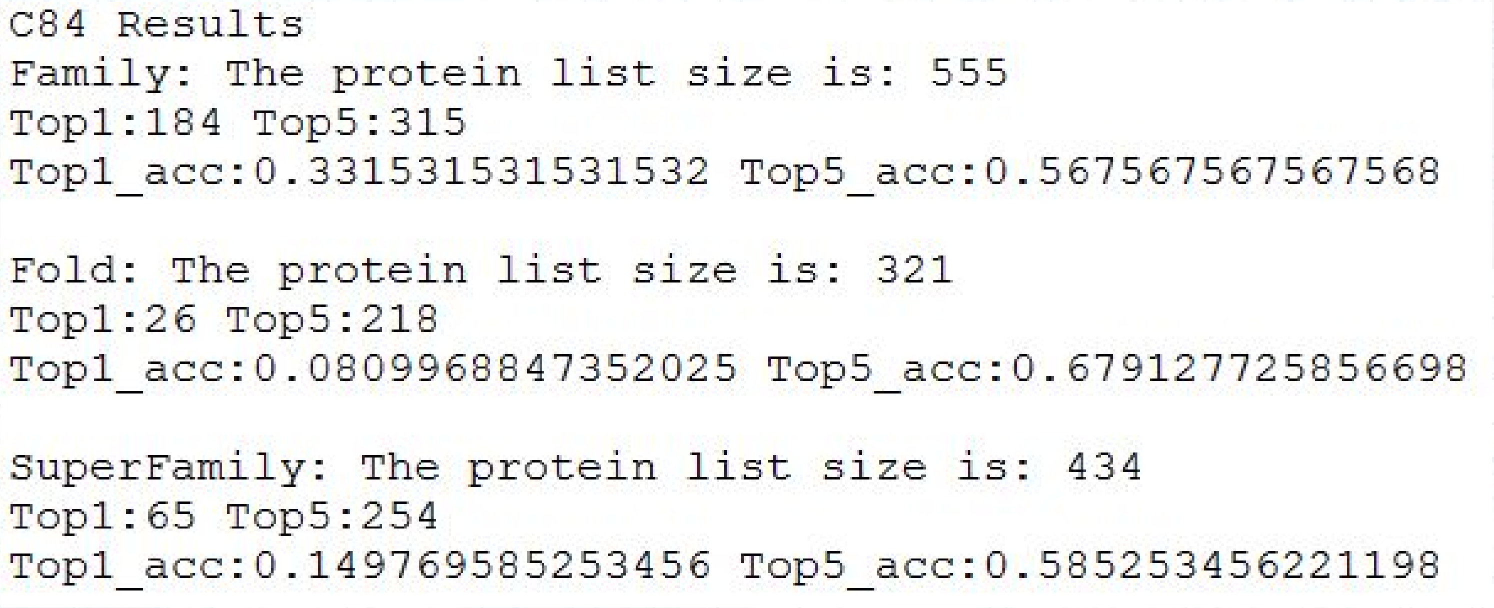
```
Examples labeled as 0 classified by model as 0: 38582 times
Examples labeled as 0 classified by model as 1: 56221 times
Examples labeled as 1 classified by model as 0: 164 times
Examples labeled as 1 classified by model as 1: 583 times

==========================Scores========================================
 Accuracy:  0.4099
 Precision: 0.503
 Recall:    0.5937
 F1 Score:  0.5446
========================================================================
```

# CNN-Fold Demo



```
tyler@ubuntu6:~/Desktop$ ls
C84K4S2.bin                cnn-fold.jar          test1.txt
C84K4S2-D150-D25-O1.json  eval-arch-test1.txt  trn1.txt
tyler@ubuntu6:~/Desktop$ head -n 10 eval-arch-test1.txt
[[0.46,0.54]
 [0.48,0.52]
 [0.65,0.35]
 [0.43,0.57]
 [0.46,0.54]
 [0.46,0.54]
 [0.47,0.53]
 [0.45,0.55]
 [0.42,0.58]
 [0.44,0.56]
tyler@ubuntu6:~/Desktop$
```

# CNN-Fold Demo



```
C84 Results
Family: The protein list size is: 555
Top1:184 Top5:315
Top1_acc:0.331531531531532 Top5_acc:0.567567567567568

Fold: The protein list size is: 321
Top1:26 Top5:218
Top1_acc:0.0809968847352025 Top5_acc:0.679127725856698

SuperFamily: The protein list size is: 434
Top1:65 Top5:254
Top1_acc:0.149769585253456 Top5_acc:0.585253456221198
```

# Future Work

- Additional methods of conveying proteins
  - Mimic image or sound
  - Data normalization
- CNN-Fold
  - Generalized input
  - Additional command line parameters

Thank you!

# Image Reference

- Protein - https://commons.wikimedia.org/wiki/File:Protein_HSPA8_PDB_1atr.png
- X Ray - http://chemwiki.ucdavis.edu/Core/Analytical_Chemistry/Instrumental_Analysis/Diffraction/X-ray_Crystallography
- NMR - http://www.nature.com/horizon/proteinfolding/background/figs/technology_f3.html
- DNN - Jo, Taeho, et al. "Improving Protein Fold Recognition by Deep Learning Networks." Scientific reports 5 (2015).
- Filter - http://eric-yuan.me/fake-cnn/
- Random data- http://www.thesearchagents.com/2013/06/google-wants-to-collect-our-data-to-show-us-what-we-want-why-do-our-governments-want-it/
- Convolution gif - https://upload.wikimedia.org/wikipedia/commons/b/b9/Convolution_of_spiky_function_with_box2.gif
- Downsampling – deeplearning4j.org
- Dropout - Srivastava et. al.