

### **SEARCH**

Change the convolutional activation function to each of these options.

## Values

"relu", "selu",
"elu", "tanh"



### **SEARCH**

Vary the learning rate and batch size of the standard convolutional network.

### Values

Learning rate: 0.01, 0.001, 0.0001, 0.00001 Batch Size: 128, 256, 512, 1024, 2048



#### SEARCH

Change the number of convolutional filters in the first layer. Double the number of filters in each subsequent layer.

#### Values

Filters: 8, 16, 32, 48, 64, 128



#### **SEARCH**

Use two Dense hidden layers before the output Dense layer. Vary the neuron count.

## Values

First Dense Neuron Count: 128, 256, 512 Second Dense Neuron Count: 512, 256, 128



#### **SEARCH**

Compare
Max Pooling,
Average Pooling,
and Strided
Convolutions
for spatial
dimensionality
reduction.

### Values

MaxPooling2D, AveragePooling2D, Conv2D(..., stride=(2,2))



### **SEARCH**

Retrain the network with 1 year's worth of examples removed from the training data.

#### Values

 $valid\_dates.year$  != 2011, 2012, 2013, 2014, 2015



#### **SEARCH**

Use 2 convolutional layers between each pooling layer. Vary the ratio of the number of filters used between the first and second layer.

# Values

same, 2x first, 0.5x first



#### SEARCH

Use SpatialDropout2D layers after each convolutional layer and vary the dropout rate.

### Values

0.1, 0.2, 0.3, 0.4, 0.5



## SEARCH

Place
BatchNormalization
layers after each
convolutional
layer. Vary
the momentum
parameter.

## Values

momentum=0.5, 0.9, 0.99, 0.999



## **SEARCH**

Add 12 kernel regularizers to each Conv2D and Dense layer. Vary the 12 strength.

### Values

strength=0.1, 0.01, 0.001, 0.0001



# SEARCH

Change the optimizer. Use the same learning rate and batch size for each.

# Values

SGD, Adam, RMSprop, Adadelta, Nadam



#### SEARCH

Vary the number of Conv2D-Activation-MaxPooling2D layer sets.

# Values

1, 2, 3, 4



## **SEARCH**

Use 2 input fields instead of 3. Vary the combinations of inputs used.

### Values

(refl, u), (refl, v), (u, v)



#### SEARCH

Change the kernel\_initializer and bias\_initializer for each Conv2D and Dense layer.

# Values

glorot\_uniform, he\_uniform, lecun\_uniform, he\_normal



#### **SEARCH**

Vary the width of the convolutional filters in each layer.

### Values

(3,3), (5,5), (7,7), (9,9)



## **SEARCH**

Rescale the spatial dimension of the input with Average Pooling 2D or Upsampling 2D layers after the input layer.

### Values

AveragePooling2D: size=(2,2), (4,4) Upsampling2D: size=(2,2), (4,4)



## SEARCH

Replace the Conv2D layers with SeparableConv2D layers. Vary the depth\_multiplier parameter.

## Values

depth\_multiplier= 1, 2, 4, 8

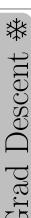


# SEARCH

Use GaussianDropout layers after each Conv2D layer. Vary the dropout rate.

### Values

rate=0.1, 0.2, 0.3, 0.4, 0.5



#### **SEARCH**

Add 11 kernel regularizers to each Conv2D and Dense layer. Vary the 11 strength.

#### Values

strength=0.1, 0.01, 0.001, 0.0001



## SEARCH

Use the SGD optimizer and vary the learning\_rate and momentum parameters.

### Values

learning\_rate=0.1, 0.01, 0.001, 0.0001 momentum=0.9, 0.99, 0.999



#### **SEARCH**

Vary the number of Conv2D-Activation-AveragePooling2D layer sets.

#### Values

1, 2, 3, 4



# SEARCH

Use 1 input field instead of 3.

# Values

refl, u, v



#### SEARCH

Change how the input values are scaled.  $scaled = a + \frac{x - x_{min}(b-a)}{x_{max} - x_{min}}$ 

### Values

Min-Max Scaling from a=0 to b=1, Min-Max Scaling from a=-1 to b=1.



#### **SEARCH**

After each pooling layer, use 2 parallel Conv2D layers with different widths and Concatenate them afterward.

### Values

(3, 3) and (5,5); (1, 1) and (3, 3)



#### **SEARCH**

Place a Flatten layer after the input and replace all Conv2D and Pooling layers with 3 Dense layers with ReLU. Vary the hidden neuron count.

### Values

(512, 256, 128), (512, 512, 512), (128, 256, 512)



## SEARCH

Remove all pooling layers. Use 4 convolution filters where the filter width doubles with each layer.

## Values

start filter width=(3, 3), (5, 5), (7, 7)



## SEARCH

Use GaussianNoise layers after each Conv2D layer. Vary the noise standard deviation.

### Values

 $\begin{array}{l} {\rm standard} \\ {\rm deviation}{=}0.1, \\ 0.01,\, 0.001 \end{array}$ 

0