Grad Descent ❄

SEARCH

Change the convolutional activation function to each of these options.

Values

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

Grad Descent ❄

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

Grad Descent ❄

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

Grad Descent ❄

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

Grad Descent ❄

SEARCH

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

Values

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

Grad Descent ❄

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

Grad Descent ❄

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

Grad Descent ❄

SEARCH

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

Values

0.1, 0.2, 0.3, 0.4, 0.5

Grad Descent ❄

SEARCH

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

Values

momentum=0.5, 0.9, 0.99, 0.999

Grad Descent ❄

SEARCH

Add l2 kernel regularizers to each Conv2D and Dense layer. Vary the l2 strength.

Values

strength=0.1, 0.01, 0.001, 0.0001

Grad Descent ❄

SEARCH

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

Values

SGD, Adam, RMSprop, Adadelta, Nadam

Grad Descent ❄

SEARCH

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

Values

1, 2, 3, 4

Grad Descent ❄

SEARCH

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

Values

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

Grad Descent ❄

SEARCH

Change the *kernel_initializer* and *bias_initializer* for each Conv2D and Dense layer.

Values

glorot_uniform, he_uniform, lecun_uniform, he_normal

Grad Descent ❄

SEARCH

Vary the width of the convolutional filters in each layer.

Values

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

Grad Descent ❄

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)

Grad Descent ❄

SEARCH

Replace the Conv2D layers with SeparableConv2D layers. Vary the depth_multiplier parameter.

Values

depth_multiplier= 1, 2, 4, 8

Grad Descent ❄

SEARCH

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

Values

rate=0.1, 0.2, 0.3, 0.4, 0.5

Grad Descent ❄

SEARCH

Add l1 kernel regularizers to each Conv2D and Dense layer. Vary the l1 strength.

Values

strength=0.1, 0.01, 0.001, 0.0001

Grad Descent ❄

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

Grad Descent ❄

SEARCH

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

Values

1, 2, 3, 4

Grad Descent ❄

SEARCH

Use 1 input field instead of 3.

Values

refl, u, v

Grad Descent ❄

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.

Grad Descent ❄

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)

Grad Descent ❄

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)

Grad Descent ❄

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)

Grad Descent ❄

SEARCH

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

Values

standard deviation=0.1, 0.01, 0.001