

Mål sektion A

Koda ihop ett Convolutional Neural Network (CNN) som analyserar MNIST Fashion-data med hjälp av Keras. En metod som ger bättre resultat än ett alla-till-alla nät ska skapas. Minst två olika inställningar på någon av parametrarna (strides, fönsterstorlek eller padding) ska köras och jämföras. Minst 90% rätt på testdatan ska uppnås.

Metod sektion A

Källkod: lab1a.py

Mitt CNN var inspirerat av (6), där ett CNN var skapat med ett första lager Conv2D, följt av ett lager MaxPooling2D, följt av två till par av lager av samma typ med ökande filterstorlek. Mitt CNN nätverk var sedan tillplattat med ett Flatten-lager, följt av ett 128-enhets Dense-lager följt av ett 10-enhets stort Dense-lager där man får ut sannolikheterna för de tio kategorierna av Fashion-datan. Olika kombinationer av inställningar testades sedan och jämfördes mot varandra. För mer information se källkoden.

Resultat sektion A

Resultatet efter att ha kört med olika inställningar visas nedan. Ett CNN med inställningarna "padding, small window, large stride" gav bäst resultat, med en noggrannhet 0,9091 på testdatan.

Inställning	Resultat
Fully connected	0.8837
No padding, small window, small strides	0.8854
No padding, small window, large strides	0.8848
No padding, large window, small strides	0.8936
No padding, large window, large strides	0.8790
Padding, small window, small strides	0.9091
Padding, small window, large strides	0.9008
Padding, large window, small strides	0.9048
Padding, large window, large strides	0.9029

Mål sektion B

Testa att använda metoder för regularisering och även Data augmentation.

Metod sektion B

Källkod: lab1b.py

Från sektion A användes inställningen "Padding, small window, small strides", och Dropout(0.25) infördes efter varje MaxPooling2D-lager samt Dropout(0.5) användes efter det alla-till-alla nät som används i slutet. Ett till Dense(128,...)-lager lades till i slutet. En inställning för L2-regularisering testades även.

Resultat sektion B

Efter att ha testat med regularisering gav dropout bäst effekt. Inställningen för L2-regularisering gav inte bättre resultat, inte heller image augmentation gav bättre resultat. Bäst resultat gavs av att endast använda dropout med ett resultat av 0.9135.

Inställning	Resultat
Fully connected, image augmentation	0.8557
Fully connected	0.8837
Padding, small window, small strides, dropout	0.9135
Padding, small window, small strides, dropout, l2 regularizer, image augmentation	0.8959

Mål sektion C

Filtrens inlärdade detektioner ska visualiseras, om hur nätverkets indata analyseras i de olika lagren.

Metod sektion C

Source: lab1c.py

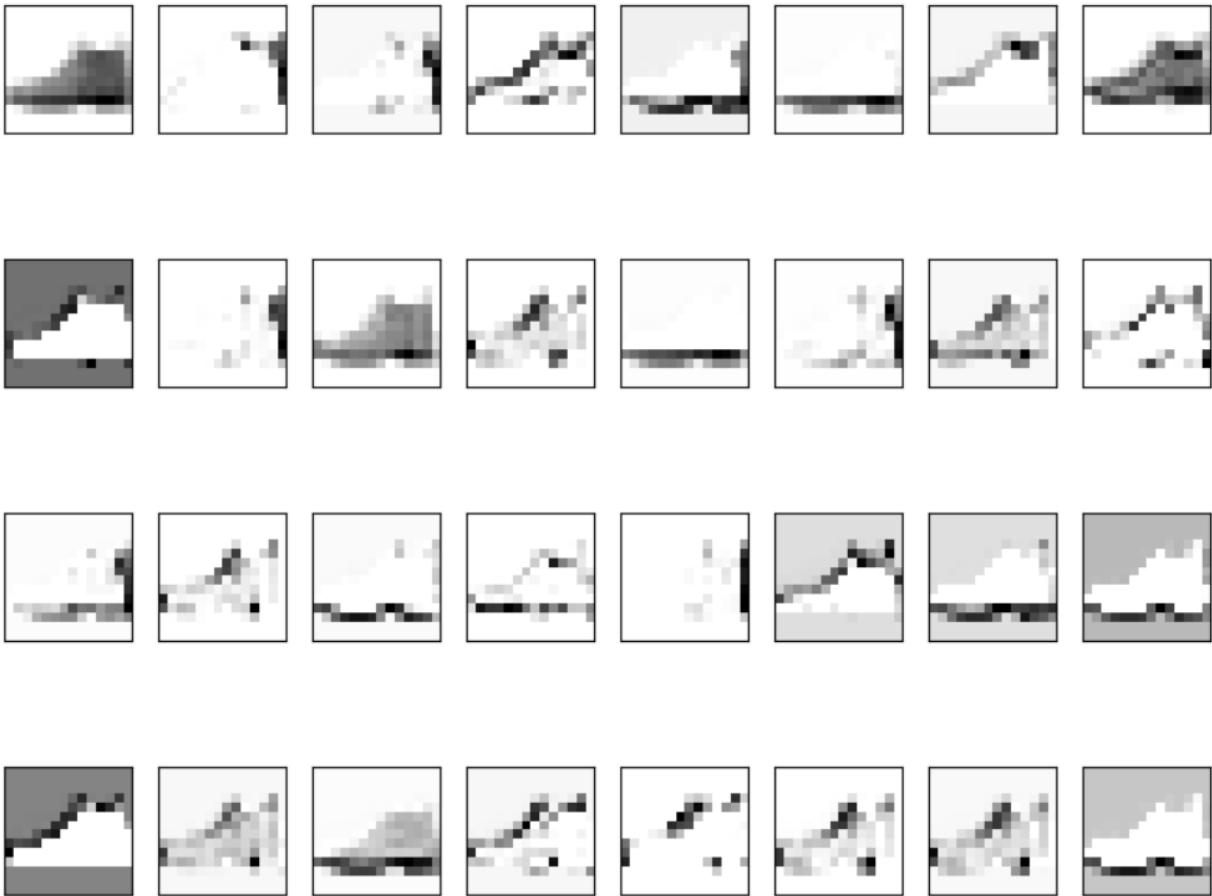
The CNN was trained on the test image set, then the model was used for prediction on a test image and the model layer predictions was extracted and the 32 first images of the prediction layers was plotted for layer 1-8 except 6.

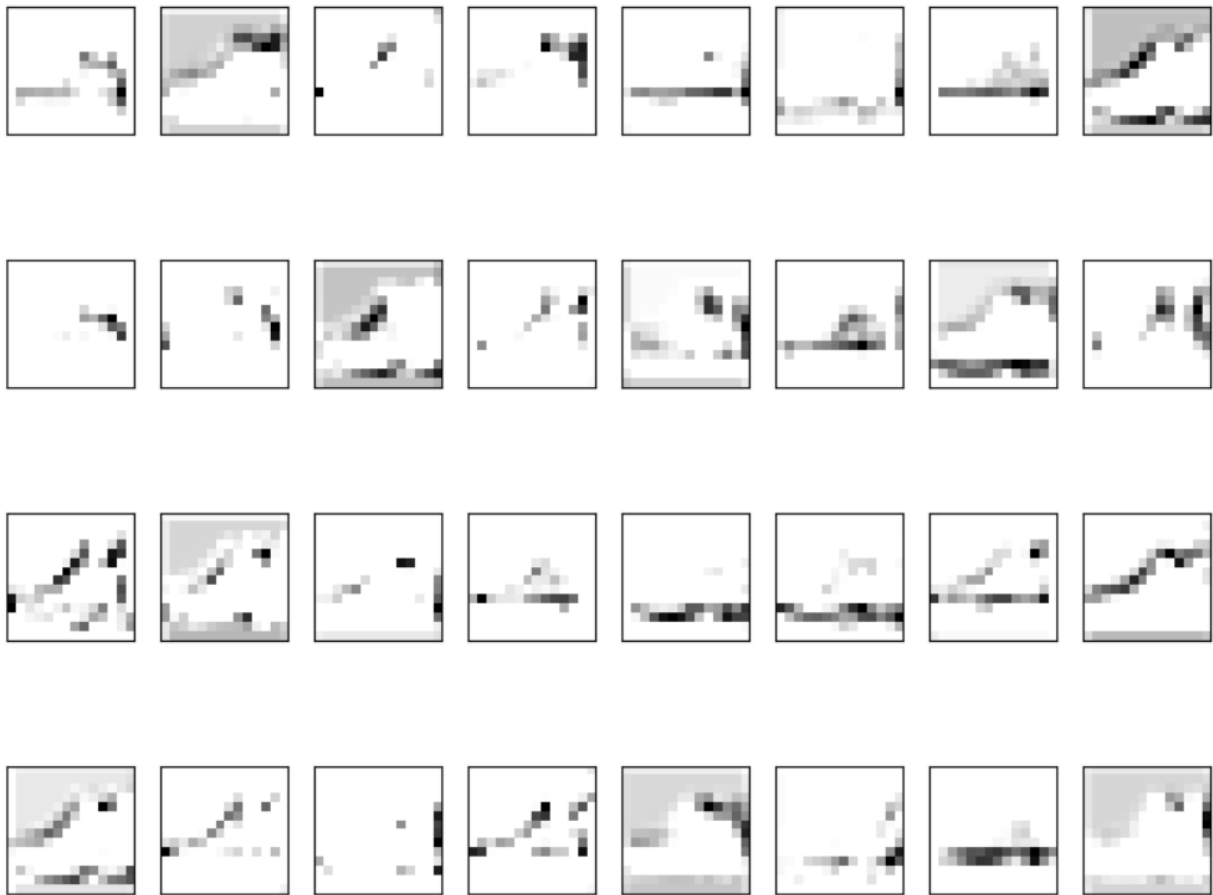
Resultat sektion C

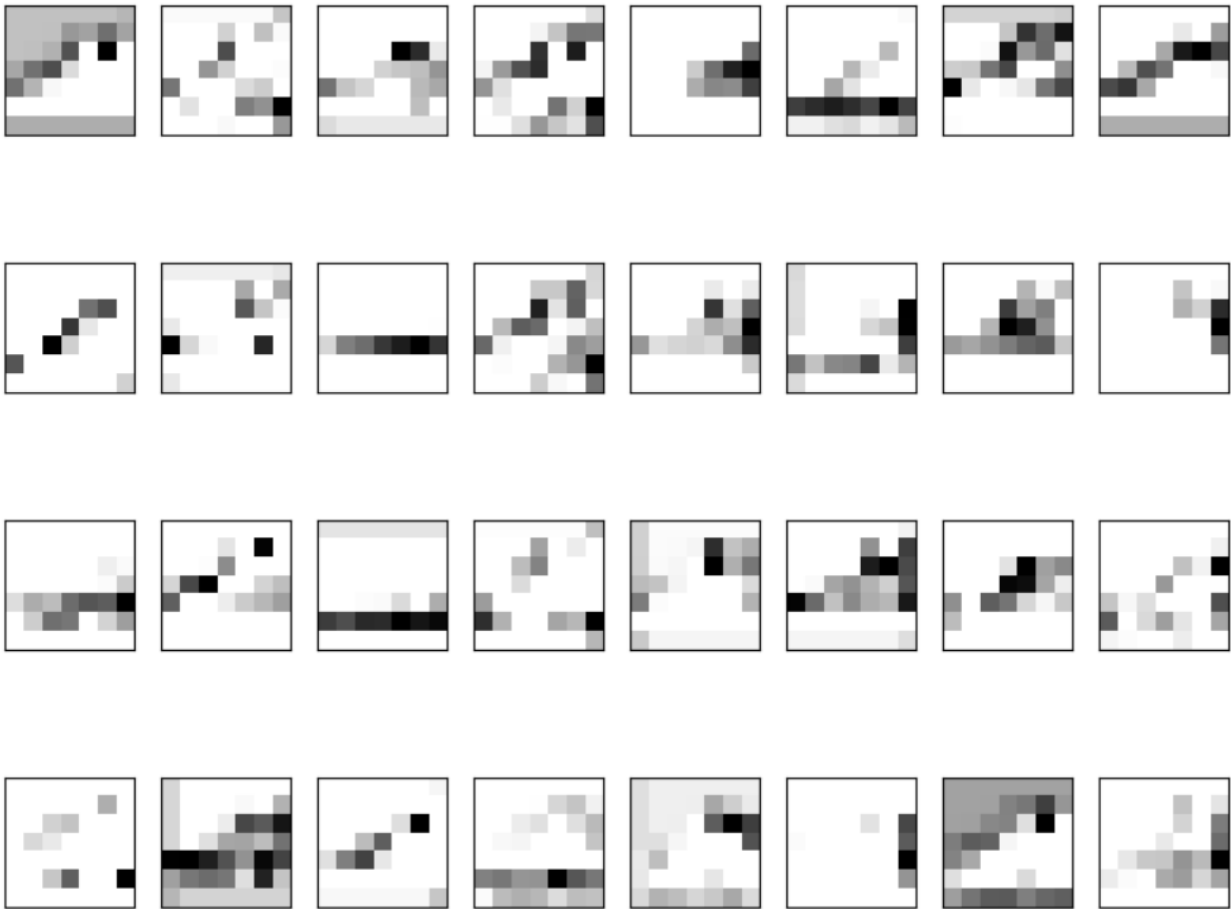
Model summary below, with the convolutional base consisting of layer 1 to 8.
Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 28, 28, 32)	320
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 32)	0
dropout_1 (Dropout)	(None, 14, 14, 32)	0
conv2d_2 (Conv2D)	(None, 14, 14, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 7, 7, 64)	0
dropout_2 (Dropout)	(None, 7, 7, 64)	0
conv2d_3 (Conv2D)	(None, 7, 7, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 3, 3, 128)	0
dropout_3 (Dropout)	(None, 3, 3, 128)	0
flatten_1 (Flatten)	(None, 1152)	0
dense_1 (Dense)	(None, 128)	147584
dense_2 (Dense)	(None, 128)	16512
dropout_4 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 10)	1290

First 32 images of layers 1, 2, 3, 4, 5, 7 and 8 are shown below:

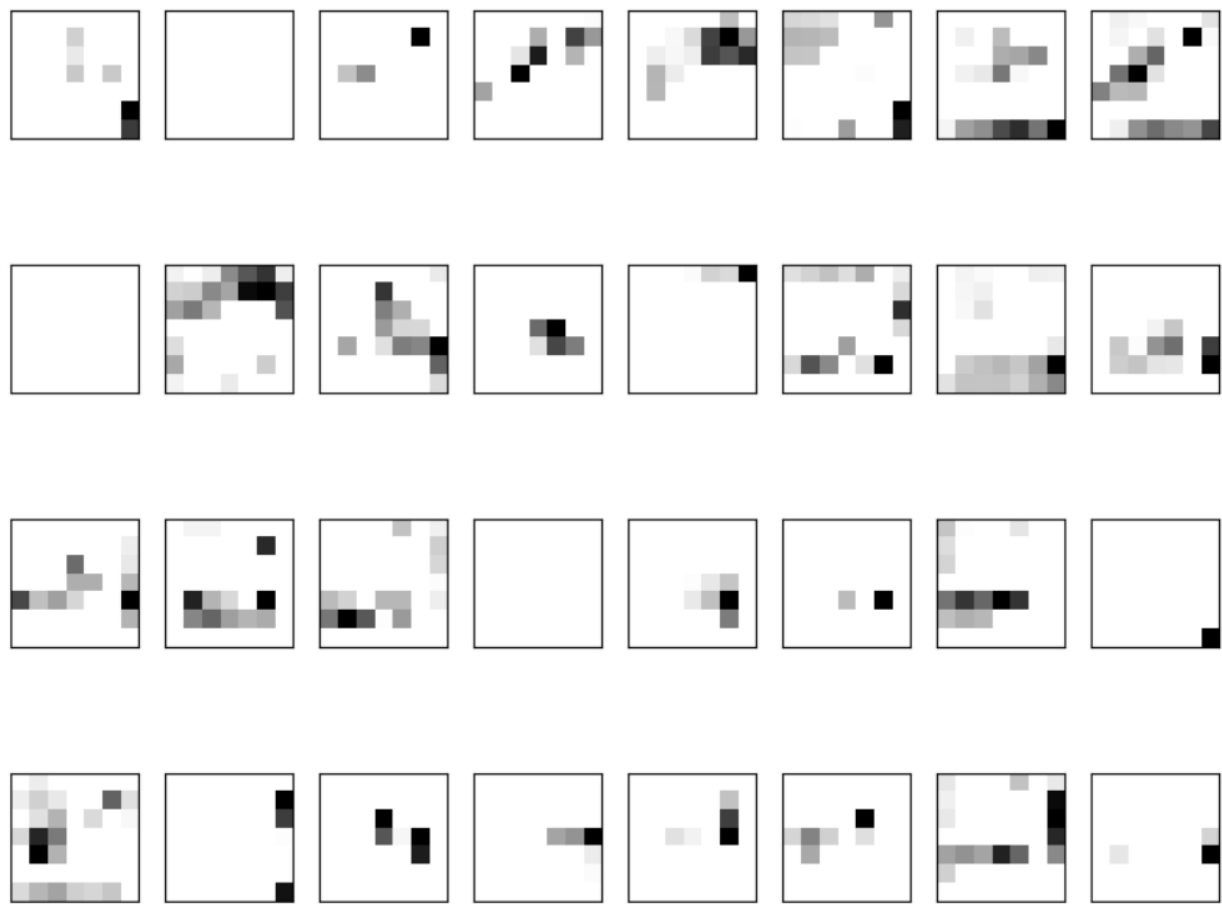


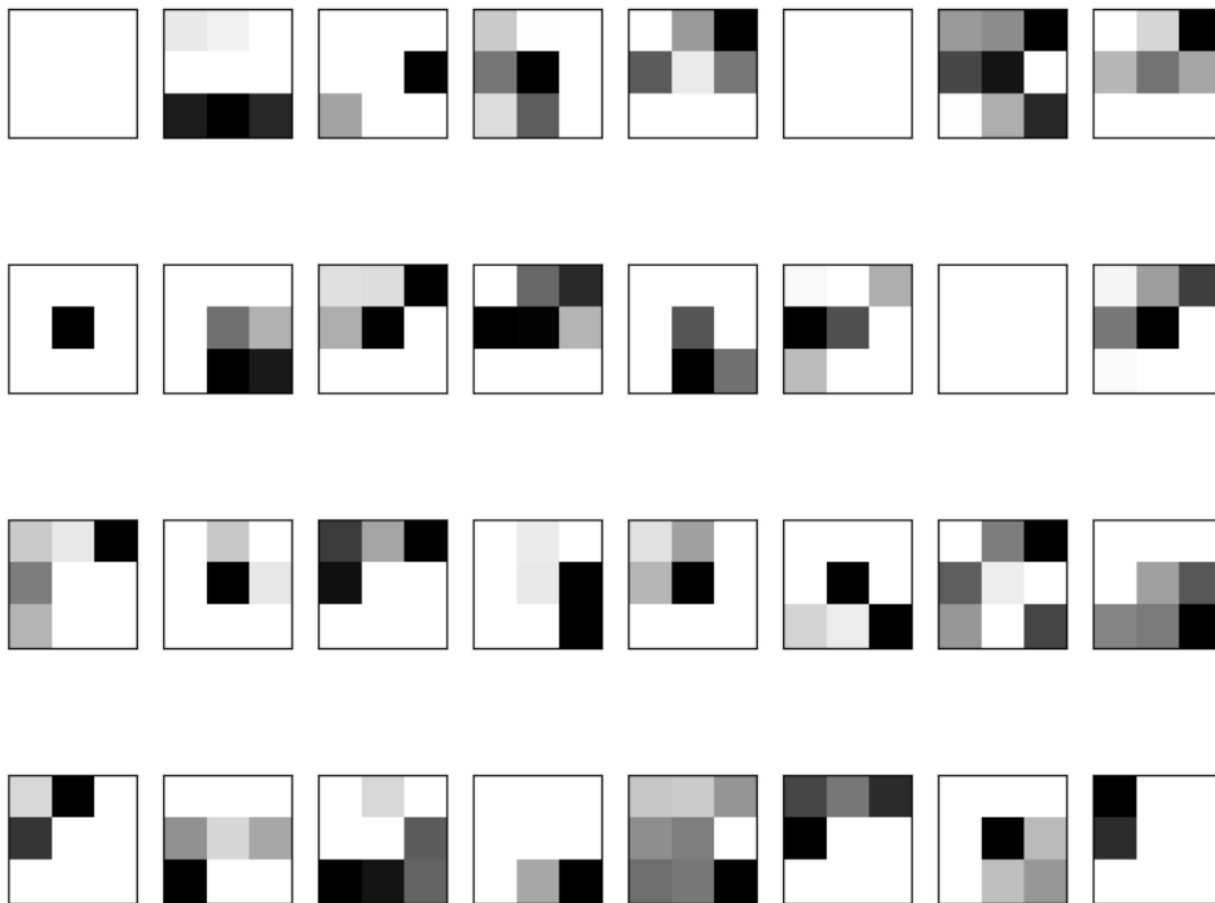




Deep Learning – laboration 1

Tony Karlsson





Referenser

- (1) DEEP LEARNING with Python, Francois Chollet
- (2) <https://keras.io/layers/convolutional/>
- (3) <https://elitedatascience.com/keras-tutorial-deep-learning-in-python>
- (4) <https://missinglink.ai/guides/keras/keras-conv2d-working-cnn-2d-convolutions-keras/>
- (5) <https://machinelearningmastery.com/padding-and-stride-for-convolutional-neural-networks/>
- (6) <https://www.pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/>