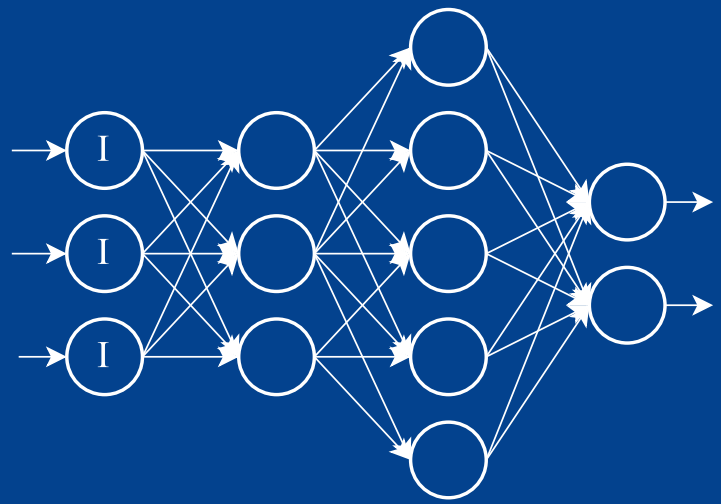
AARHUS
UNIVERSITY
SCHOOL OF ENGINEERING

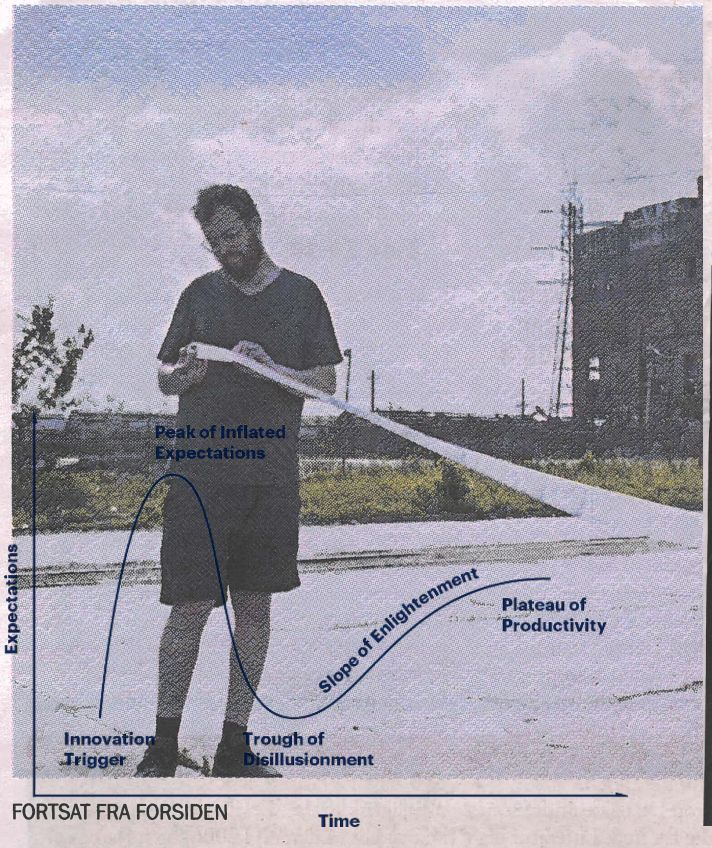
LESSON 6: Neural Networks

CARSTEN EIE FRIGAARD

AUTUMN 2021




SmartBus
Powered by holo
Hop ombord
på Danmarks første
førerløse bus.



FORTSAT FRA FORSIDEN


Weekendavisen # 01 4. januar 2019

Amos Oz. Israels store forfatter er død. Læs Martin Krasniks nekrolog og andre reaktioner. **Side 3 og 4**

Essay. Kathrine Tschemerinskys jagt på sit navn førte hende til Hviderusland. **Side 8-11**

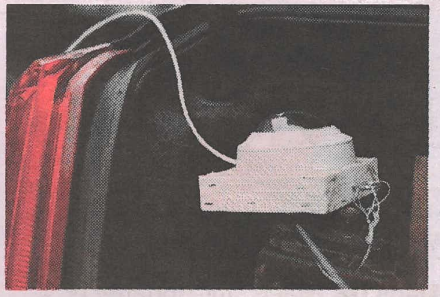
Provins-Proust. Andreas Maier er nået til fjerde bind i sin søgen efter sit udkantstyske jeg. **Side 6**

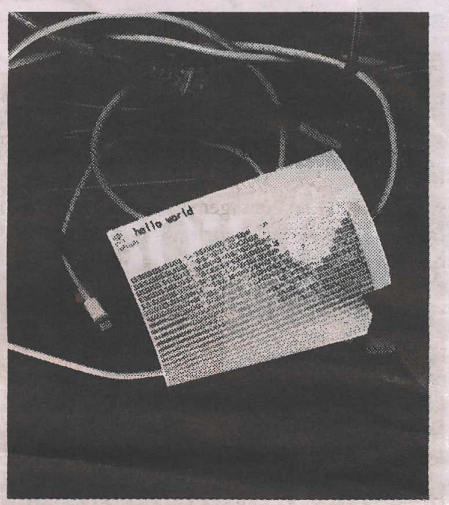
Stilskifte. Fin antologi viser, hvad der skete, da bladtegnerne blev bogillustratorer. **Side 14-15**

Bøger

Interview. Gonzo-programmøren Ross Goodwin har forfattet en kunstig intelligens, som har skrevet en surrealistisk beatroman.

På roadtrip med en insekthjerne





»Det er et forsøg på at skabe en ny brugerflade for at skrive. På en måde har jeg jo skrevet en roman med en bil,« fortæller Ross Goodwin om sin AI-forfattede bog *1 the road*.
FOTOS: CHRISTIANA CARO

På roadtrip med en insekthjerne

Af MIKKEL BORIS

I 2016 slog computeren AlphaGo den 18-dobbelte verdensmester i brætspillet Go, Lee Sedol. Go er et kompliceret og abstrakt spil, som kræver intuition og kreativitet, men den kunstige intelligens vandt med en række innovative træk overlegent. Undervejs i spillet troede kommentatorerne, at der var fejl i systemet, men trækkene viste Goodwin til Weekendavisen fra sin lejlighed i Los Angeles.

Inden køreturen havde han brugt måneder på at træne maskinen. Han satte den til at læse et stort korpus af moderne litteratur fra hele verden, så den kunne lære at skrive af de store forfattere.

»Det fungerer ligesom autokorrekturen på din telefon, bare klogere og trænet på en mere litterær kilde. Den skriver bogstav for bogstav, så den har lært sig selv at forudsige det næste tegn baseret på det foregående. Når du har

når du dekonstruerer dem. Efter at have læst den i ét stræk og fået turen lidt på afstand har romanen fået en universalitet, så jeg kan projicere mine egne oplevelser ind i teksten,« uddyber Goodwin.

– *Du har beskrevet projektet som at lære en insekthjerne at skrive. Hvad betyder det?*

»Jeg forsøgte at pointere, at maskinen ikke er på niveau med den menneskelige hjerne. Et artificielt neuralt net er en algoritme, der er lavet, som vi tror, hjerner fungerer. Jeg synes, at en insekthjerne er en god analogi, for det er


e Count: 8
e Count: 0
A T
[0 0]
0 0]
[0 0]
side
front
side
front
front

A computer vision system to monitor the infestation level of Varroa destructor in a honeybee colony

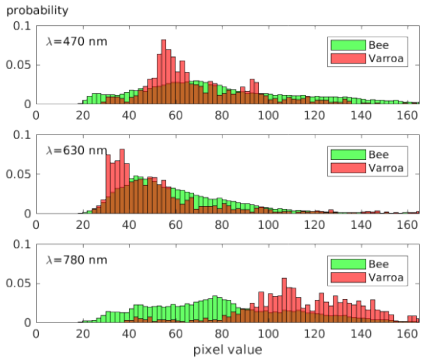


Figure 5: Histograms of bee and varroa pixel intensity values, for the spectral wavelengths 470 nm, 630 nm, and 780 nm respectively, recorded with the JAI camera. The image path is via the mirror-window-mirror, i.e. data were sampled with the setup given in figure 1. The image data for the histogram is the single bee with mite seen in figure 6.

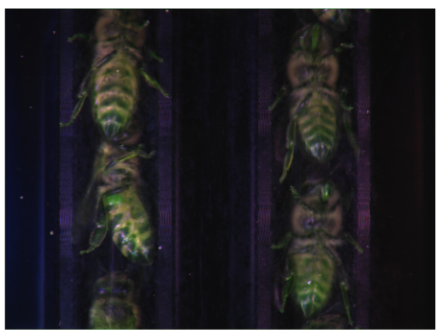


Figure 6: The actual unprocessed camera view of the bees

spectively. The CM analysis was able to rank all wavelengths combinations, using one, two, three or four district wavelengths to give a ranking list of 'best' combination also taking the JAI camera spectrum into account.

The CM value of the actual choose wavelengths combination (470-630-780 nm) gave a rank just below the CM average score. This CM analysis was conducted after picking the actual used wavelengths, so later versions of the VMU might want to investigate a CM combination with a higher rank.

A specially designed diffuser and a number of narrow spectral LE were mounted in the camera focal diffuse illuminat flections.

Figure 6 disp along the passa era, with the gre with the NIR m

2.3.3. Real-time processin

A color and tion of 1296×96 from the camer over two separat sary sustained b ing frames real-t

These data w line post-proces first matching t rally coalescing producing a 24 the later image

Lossless real-t can be applied bandwidth than

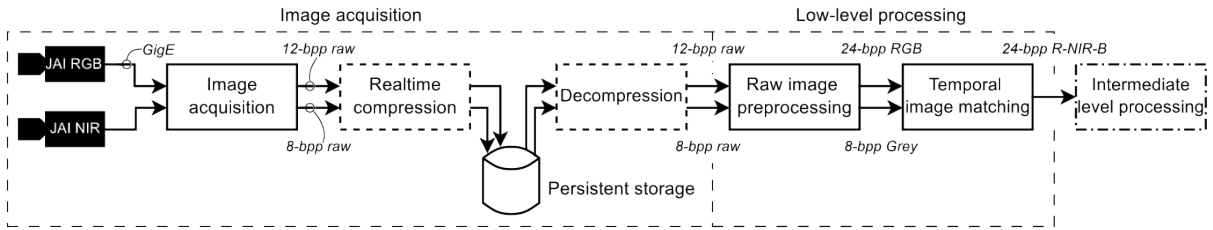


Figure 7: The low-level image processing pipeline. Raw camera images are stored on disk for later retrieval and post-processing. 12- and 8 bits-per-pixel are used as the raw JAI/Bayer packed pixel format for the RGB and IR images respectively. Lossless, real-time compression can be introduced if persistent storage bandwidth is less than the raw-stream image rate of 93 MiB/sec. The 12- and 8-bpp raw images from the network arrives out-of-order with respect to each other, hence the need for the temporal image matching.

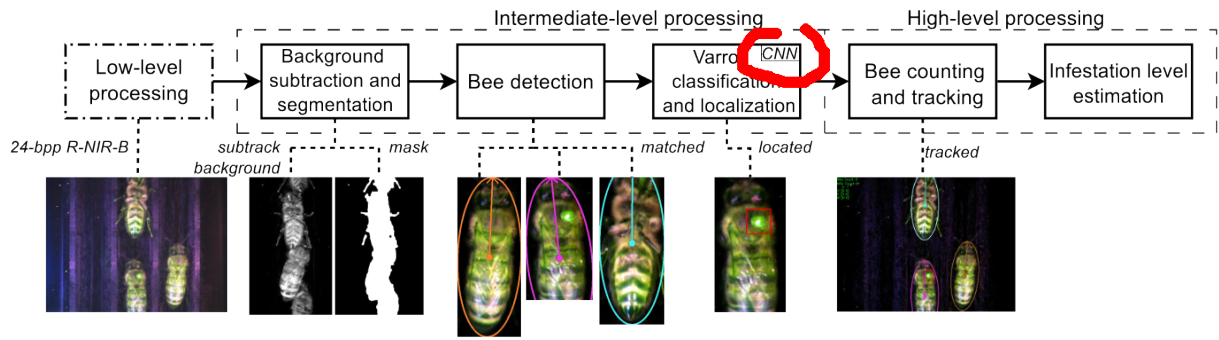
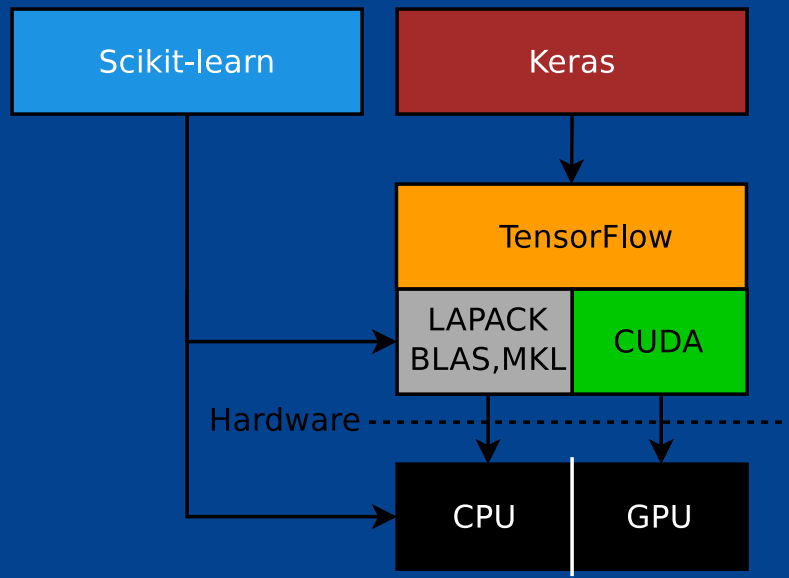


Figure 8: The processing pipeline of the intermediate- to high-level image processing algorithms to analyze and count the number of bees with *Varroa destructor*. A trained convolutional neural network (CNN) was used for the Varroa classification and localization stage.

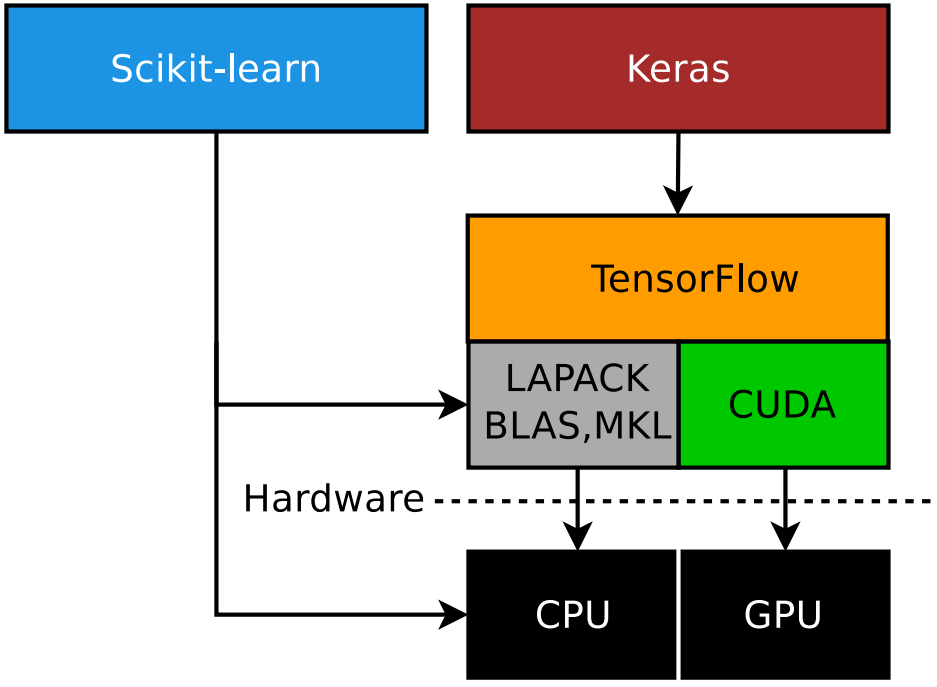
BA Project: generic tagging/labling tool

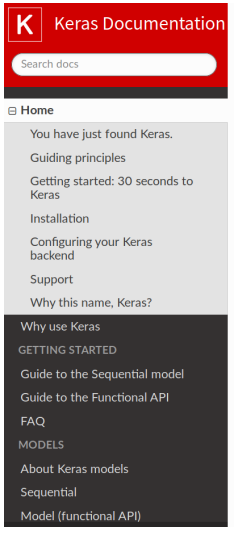
IMPLEMENTING MLPS IN KERAS



Keras and Tensorflow

Using the Keras API instead of Scikit-learn or TensorFlow



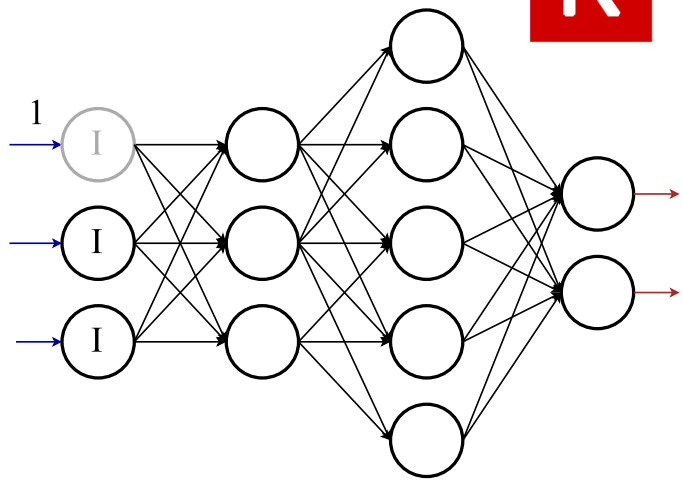


NOTE:

- documentation: `https://keras.io/`
- keras provides a `fit-predict`-interface,
- many similiarities to Scikit-learn,
- but also many differences!

Building Keras MLPs

Using the Keras `Sequential` class, programatical build up model:




```
# Build Keras model
model = Sequential()
model.add(Dense(input_dim=2, units=3, activation="tanh", ..)
model.add(Dense(units=5, activation="relu", ..)
model.add(Dense(units=2, activation="softmax"))
.
.
X_train, .. = train_test_split(X, y, .. )
.
.
y_train_categorical = to_categorical(y_train, num_classes=2)
y_test_categorical  = to_categorical(y_test,  num_classes=2)
.
.
history = model.fit(X_train, y_train_categorical, ..
.
.
score = model.evaluate(X_test, y_test_categorical)
```

Notes on Keras MLPs

Typical Keras MLP Supervised Classifier setup..

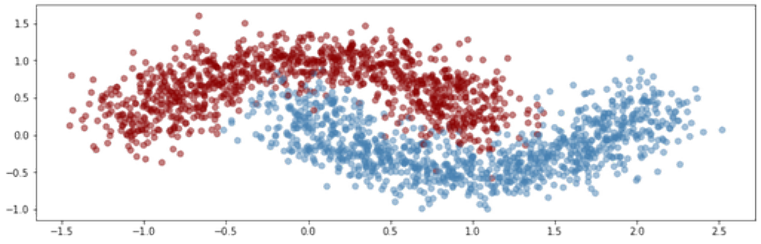
- loss function

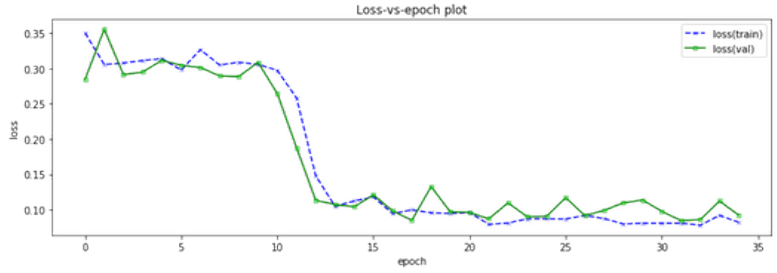
```
loss='categorical_
        crossentropy'
```

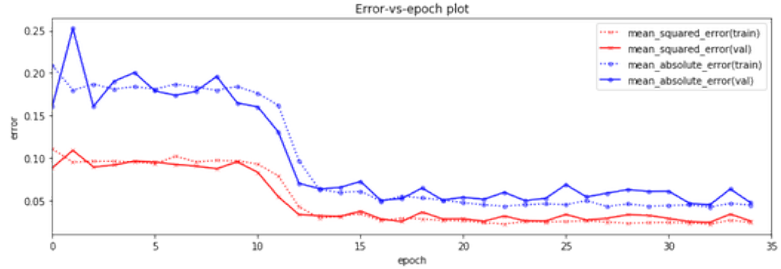
- metrics collected via history

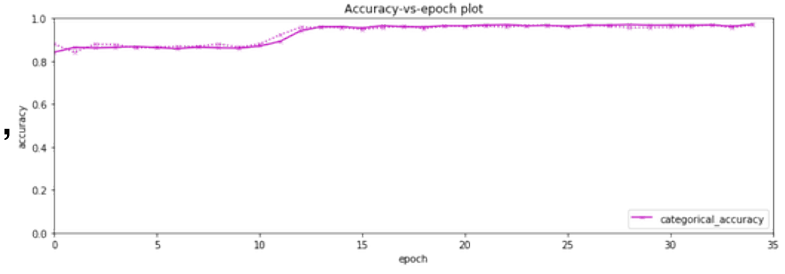
```
metrics=[
  'categorical_accuracy',
  'mean_squared_error',
  'mean_absolute_error'])
```

- input lay.: categorical encoding,
- output lay.: softmax function,









And notice that Keras do *not* provide metrics like

```
precision, recall, F1
```

but instead

```
categorical_accuracy, binary_accuracy
```

Input Layer: Categorical Encoding

For MLP Classification

One-hot `to_categorical(·)` encoding in Keras:

- input layer: one-hot class encoding,
- output layer: one neuron per output class that fires, and use `softmax` for output neurons
- beware of misformated classes.

```
1  import numpy as np
2  from keras.utils.np_utils import to_categorical
3
4  y = np.array([1, 2, 0, 4, -1])
5  y_cat = to_categorical(y)
6
7  print(y_cat)
8
9  #[[0. 1. 0. 0. 0.]  => i=0, class 1
10 # [0. 0. 1. 0. 0.]  => i=1, class 2
11 # [1. 0. 0. 0. 0.]  => i=2, class 0
12 # [0. 0. 0. 0. 1.]  => i=3, class 4
13 # [0. 0. 0. 0. 1.]] => i=4, also class 4!
14 #                      NOTE: no class 3
```

[L07/Extra/keras_to_categorical.ipynb]

Output Layer: Softmax Function

For MLP Classification: Assing a Probability for each Class

Softmax (softargmax/normalized exponential) definition

$$\text{softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{i=1}^{n} e^{x_i}}$$

`softmax`: smooth approx. of `argmax` function.

`argmax`: the index-of-the-max-value for some data.

```python
# python demo of softmax/argmax
x = np.array([1, 2, -4, 5, 1])
i = np.argmax(x)

PrintMatrix(x,"x = ")
print(f"np.argmax(x) = {np.argmax(x)}")

def softmax(x):
    z = np.exp(x)
    s = np.sum(z)
    return z / s

PrintMatrix(softmax(x),"softmax(x) = ")
print(f"np.argmax(softmax(x)) = {np.argmax(softmax(x))}")
```

```
# output
x = [ 1  2 -4  5  1 ]
np.argmax(x) = 3
softmax(x) = [0.02 0.05 0. 0.92 0.02]
np.argmax(softmax(x)) = 3
```

[L07/Extra/softmax.ipynb]

High-Performace-Computing (HPC)

Running on the ASE GPU cluster, your group login=f20malXY

NOTE:

manuel GPU hukommelses Garbage Collection...

For `keras` GPU kald:

```
StartupSequence_EnableGPU(gpu_mem_fraction=0.1, gpus=1)
```

NOTE2: script found in `/home/shared/00_init.py` that runs for all users!

EXTRA SLIDES..

MLP Effect of Number of Hidden Layers

How Many Hidden Layers?

MNIST Search Quest Exercise:

- ITMAL Grp10:
  used a 20-50-70-100-70-50-20 layer `MLPClassifier`,
- found layers by trial-and-error,
- but what are the optimal hidden layer sizes and neurons per hidden layer?

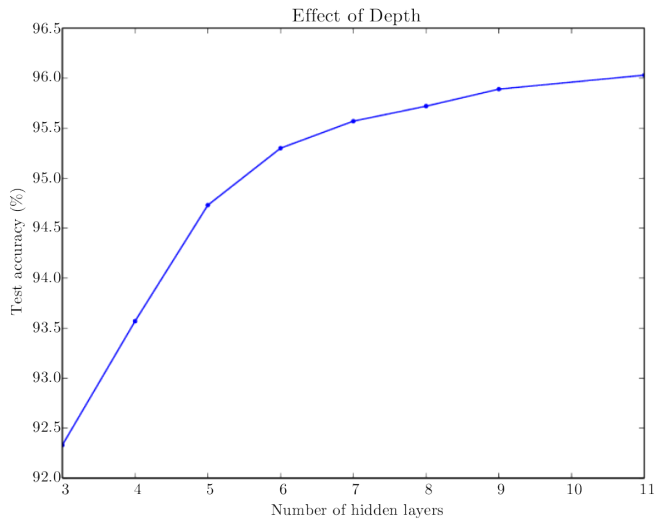


Figure 6.6: Empirical results showing that deeper networks generalize better when used to transcribe multi-digit numbers from photographs of addresses. Data from Goodfellow *et al.* (2014d). The test set accuracy consistently increases with increasing depth. See Fig. 6.7 for a control experiment demonstrating that other increases to the model size do not yield the same effect.

The ReLU Activation Function [DL]

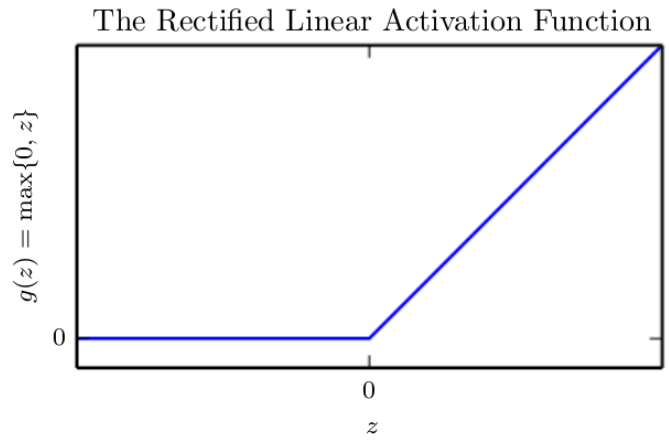


Figure 6.3: The rectified linear activation function. This activation function is the default activation function recommended for use with most feedforward neural networks. Applying this function to the output of a linear transformation yields a nonlinear transformation. However, the function remains very close to linear, in the sense that is a piecewise linear function with two linear pieces. Because rectified linear units are nearly linear, they preserve many of the properties that make linear models easy to optimize with gradient-based methods. They also preserve many of the properties that make linear models generalize well. A common principle throughout computer science is that we can build complicated systems from minimal components. Much as a Turing machine's memory needs only to be able to store 0 or 1 states, we can build a universal function approximator from rectified linear functions.

Effect of Depth [DL]

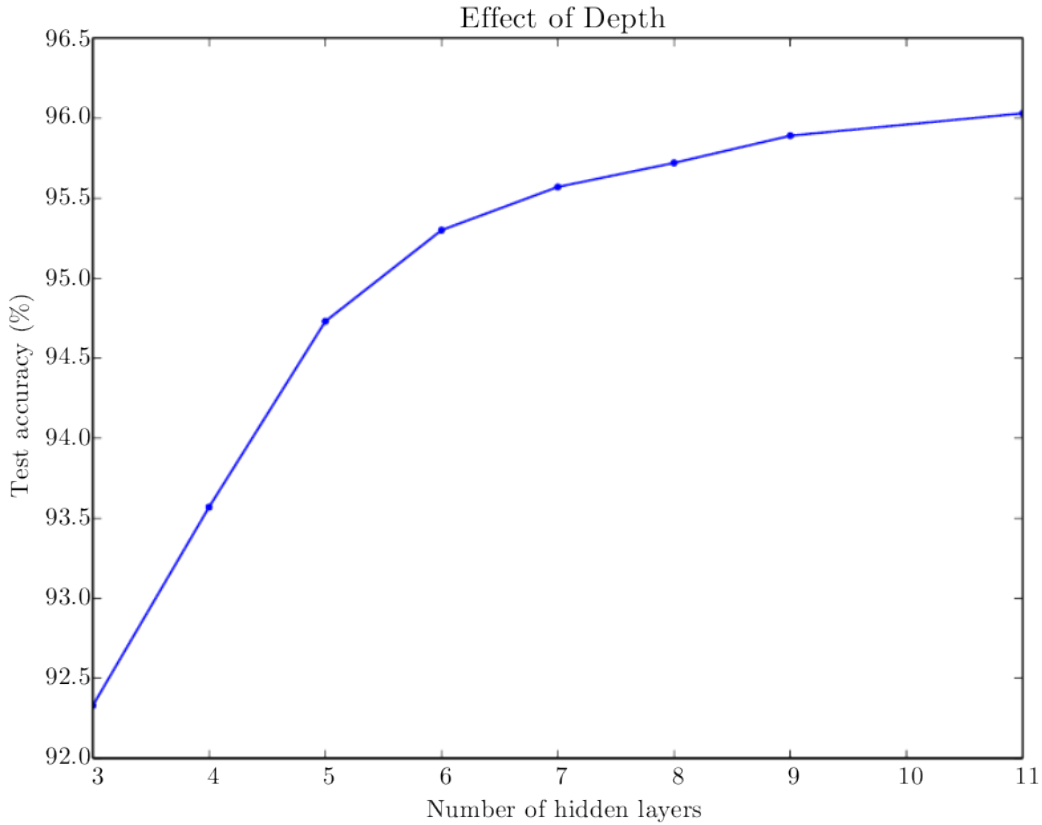


Figure 6.6: Empirical results showing that deeper networks generalize better when used to transcribe multi-digit numbers from photographs of addresses. Data from Goodfellow *et al.* (2014d). The test set accuracy consistently increases with increasing depth. See Fig. 6.7 for a control experiment demonstrating that other increases to the model size do not yield the same effect.

Effect of Number of Parameters [DL]

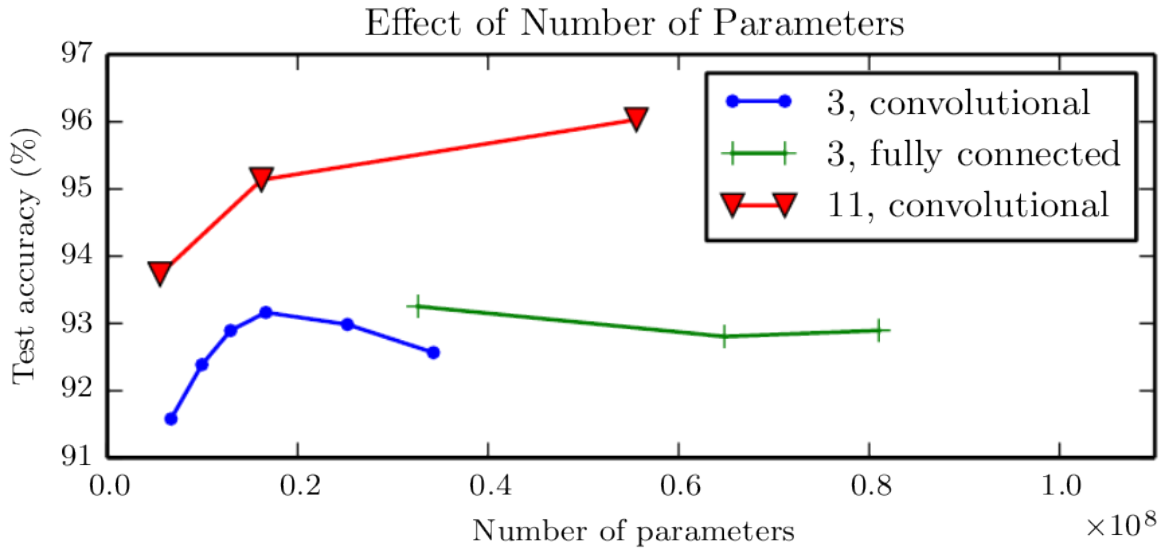


Figure 6.7: Deeper models tend to perform better. This is not merely because the model is larger. This experiment from Goodfellow *et al.* (2014d) shows that increasing the number of parameters in layers of convolutional networks without increasing their depth is not nearly as effective at increasing test set performance. The legend indicates the depth of network used to make each curve and whether the curve represents variation in the size of the convolutional or the fully connected layers. We observe that shallow models in this context overfit at around 20 million parameters while deep ones can benefit from having over 60 million. This suggests that using a deep model expresses a useful preference over the space of functions the model can learn. Specifically, it expresses a belief that the function should consist of many simpler functions composed together. This could result either in learning a representation that is composed in turn of simpler representations (e.g., corners defined in terms of edges) or in learning a program with sequentially dependent steps (e.g., first locate a set of objects, then segment them from each other, then recognize them).

Deep feedforward networks [DL]

Deep feedforward networks, also often called feedforward neural networks, or multi- layer perceptrons (MLPs), are the quintessential deep learning models.

[DL, p167]