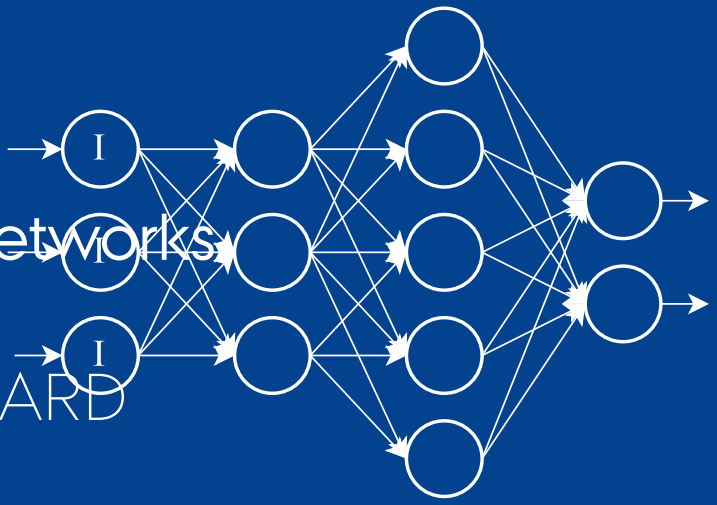
AARHUS
UNIVERSITY
SCHOOL OF ENGINEERING

LESSON 7: Neural Networks

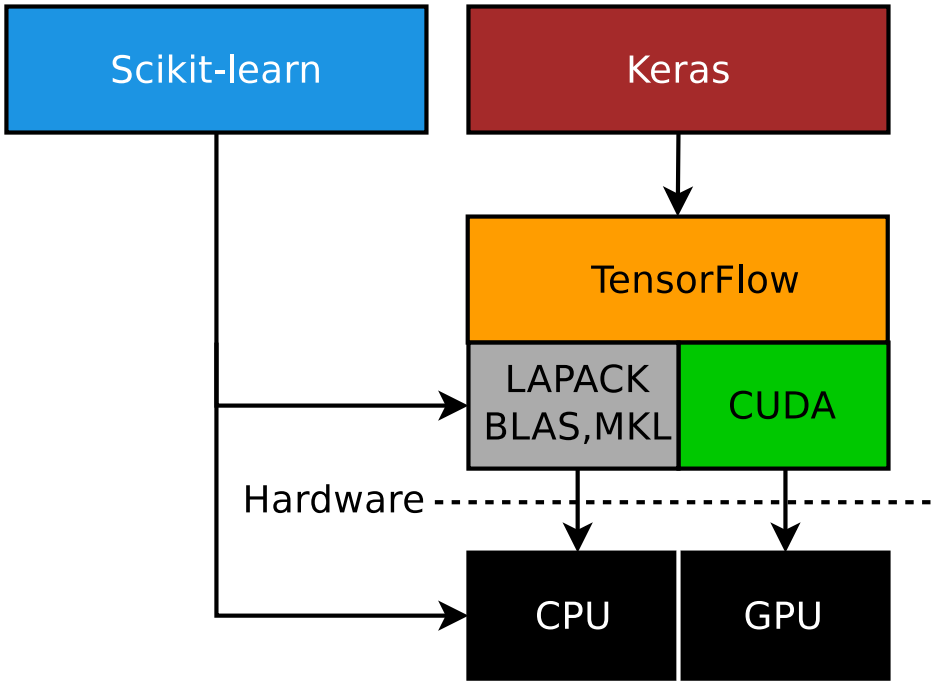


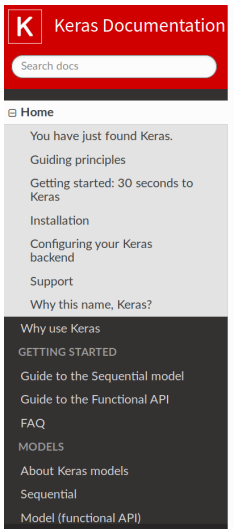
CARSTEN EIE FRIGAARD

AUTUMN 2021

Keras and Tensorflow

Using the Keras API instead of Scikit-learn or TensorFlow



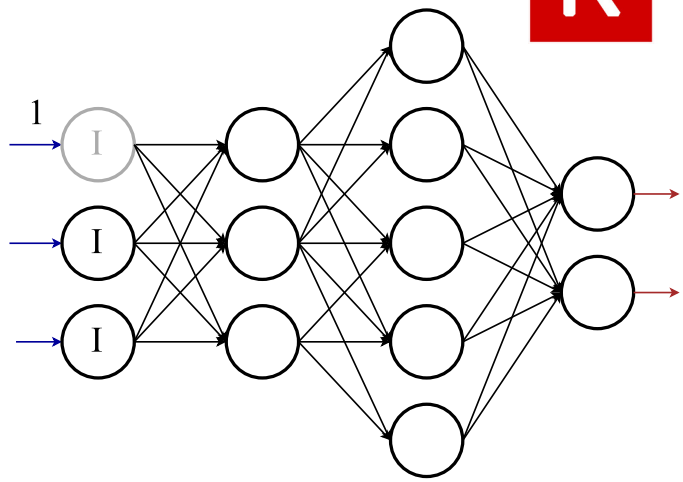


NOTE:

- 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..

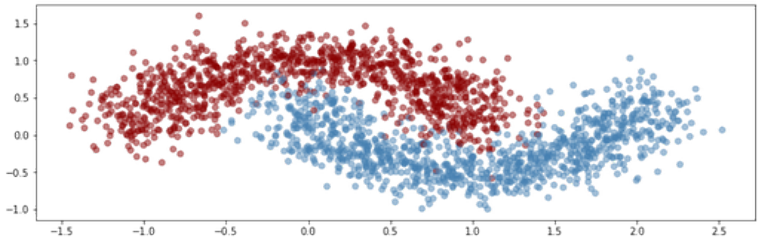
- loss function

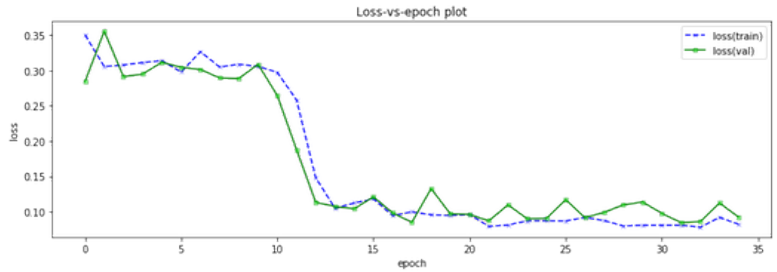
```
loss='categorical_
        crossentropy'
```

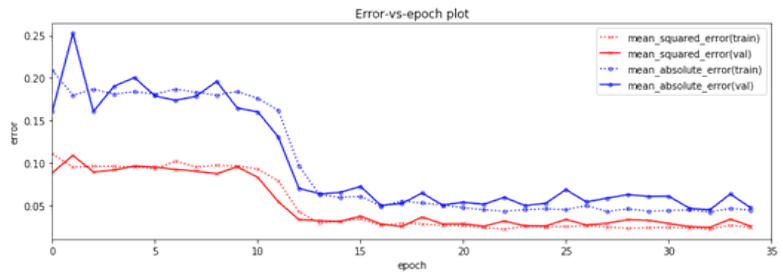
- metrics collected via history

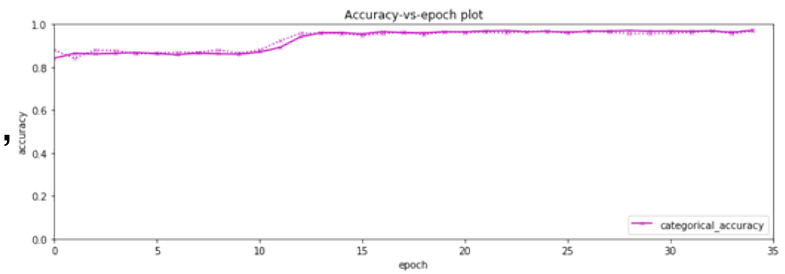
```
metrics=[
  'categorical_accuracy',
  'mean_squared_error',
  'mean_absolute_error'])
```

- 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.

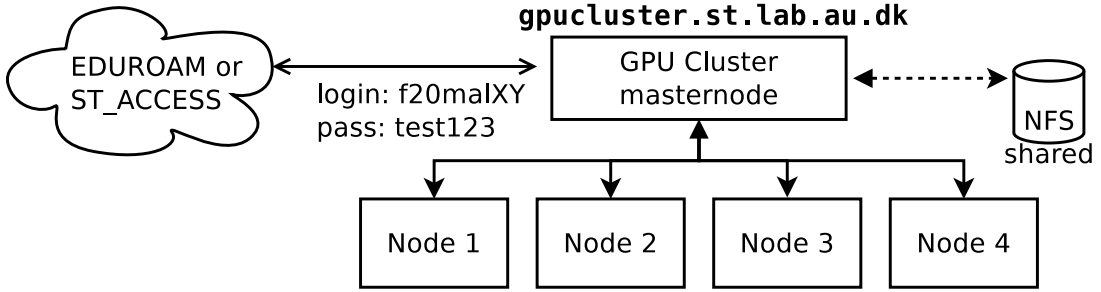
```
# python demo of softmax/argmax
x = np.array([1, 2, -4, 5, 1])
i = np.argmax(x)

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))}")
```

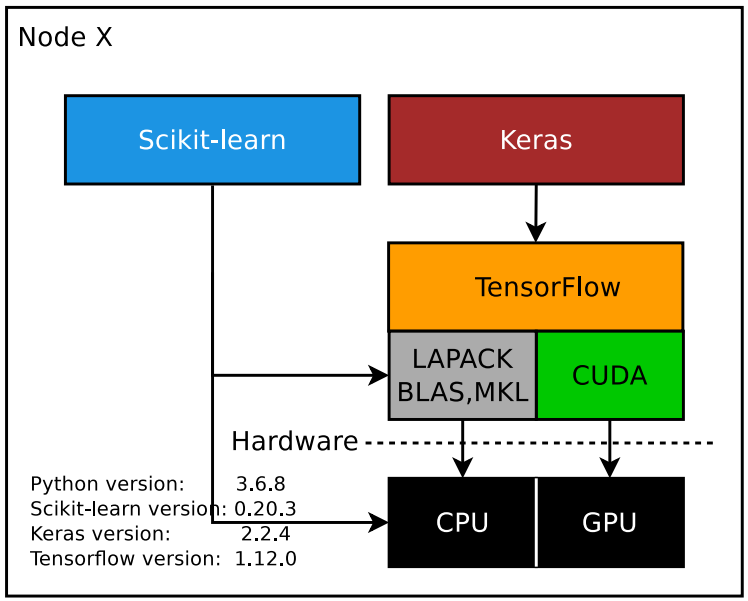
```
# output
x = [ 1  2 -4  5  1 ]
np.argmax(x) = 3
softmax(x) = [0.02 0.05 0. 0.92 0.02]
np.argmax(softmax(x)) = 3
```

[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!