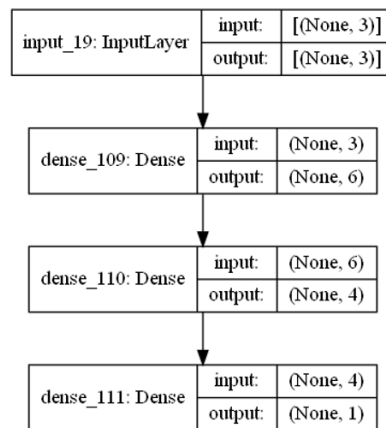


Neurale Netwerken- Oplossingen

Vraag 1 – The Simpsons Revisited

-
- binare classificatie
- name* -> nominale variabele, kan dus niet worden gebruikt, *gender* wordt gebruikt als resultaat (output), dus kan deze variabele niet gebruikt worden als input.
- Vervang de 'M' en 'F' waarden in de output kolom (*y_simpsons*) door 1 en 2. *y_simpsons* dient niet genormaliseerd te worden.
- Advies mbt de parameters (binare classificatie):
 gebruik *activation='sigmoid'* in de Dense-functies voor de input en de hidden layer(s)
 gebruik *activation='softmax'* in de Dense-functie voor de output layer
 gebruik *optimizer=Adam(learning_rate=lr)* in de compile-functie
 gebruik *loss=keras.losses.binary_crossentropy* in de compile-functie (opmerking: gaat samen met slechts een output neuron)
 gebruik *metrics= ['accuracy']* in de compile-functie
Advies mbt de nodes: gebruik verschillende hidden layers:



-

Vraag 2 – Forecast demo Revisited

-
- regressie
- De eerste 2 kolommen kunnen als input gebruikt worden, de derde kolom als output.
- Alleen de output kolom dient genormaliseerd te worden.
- Advies mbt de parameters (regressie):
 gebruik *activation='relu'* in de Dense-functies voor de input en hidden layer(s)
 gebruik *activation='linear'* in de Dense-functie voor de output layer

gebruik `optimizer=RMSprop(learning_rate=lr)` of `Adam(learning_rate=lr)` in the compile-functie

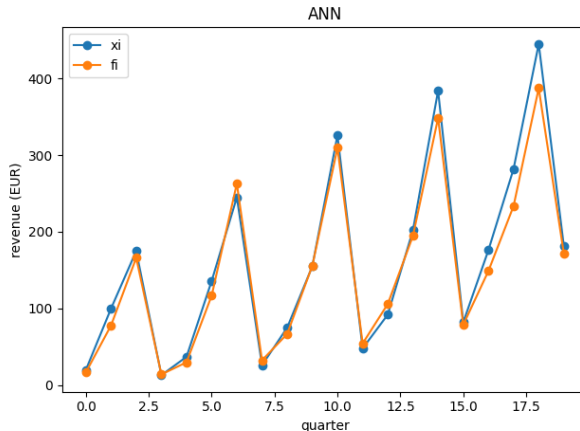
gebruik `loss=keras.losses.MeanAbsoluteError()` in the compile-functie

gebruik `metrics=keras.metrics.MeanAbsolutePercentageError()` in de compile-functie

gebruik `epochs=1000` in de fit-functie

Advies mbt de nodes: gebruik verschillende hidden layers (vb. 6 hidden layers met als aantal neuronen: 4, 8, 16, 8, 4, 2)

f.



Vraag 3 – Iris

a. -

b. multi-class classificatie

c. De eerste kolommen kunnen als input worden gebruikt, de laatste kolom bevat de output.

d. Vervang de 3 output iris-types door numerieke waarden (1, 2 en 3). Gebruik de `to_categorical` functie om de y-iris output data set te bekomen.

e. -

f. Advies mbt de parameters (multi-class classificatie):

gebruik `activation='sigmoid'` in de Dense-functies voor de input en hidden layer(s)

gebruik `activation='softmax'` in de Dense-function voor de output layer

gebruik `optimizer=Adam(learning_rate=lr)` in the compile-functie

gebruik `loss=keras.losses.categorical_crossentropy` in the compile-functie

gebruik `epochs=1000` in de fit-functioe

Advies mbt de nodes: gebruik verschillende hidden layers (vb. 4 hidden layers met als aantal neuronen: 4, 8, 16, 8) en hou rekening met het aantal categorieën in de output data set

g. Accuracy zeer dicht bij 1