

22.12.20

***Einführung in Data  
Science und maschinelles  
Lernen mit R***

**Neuronale Netze**

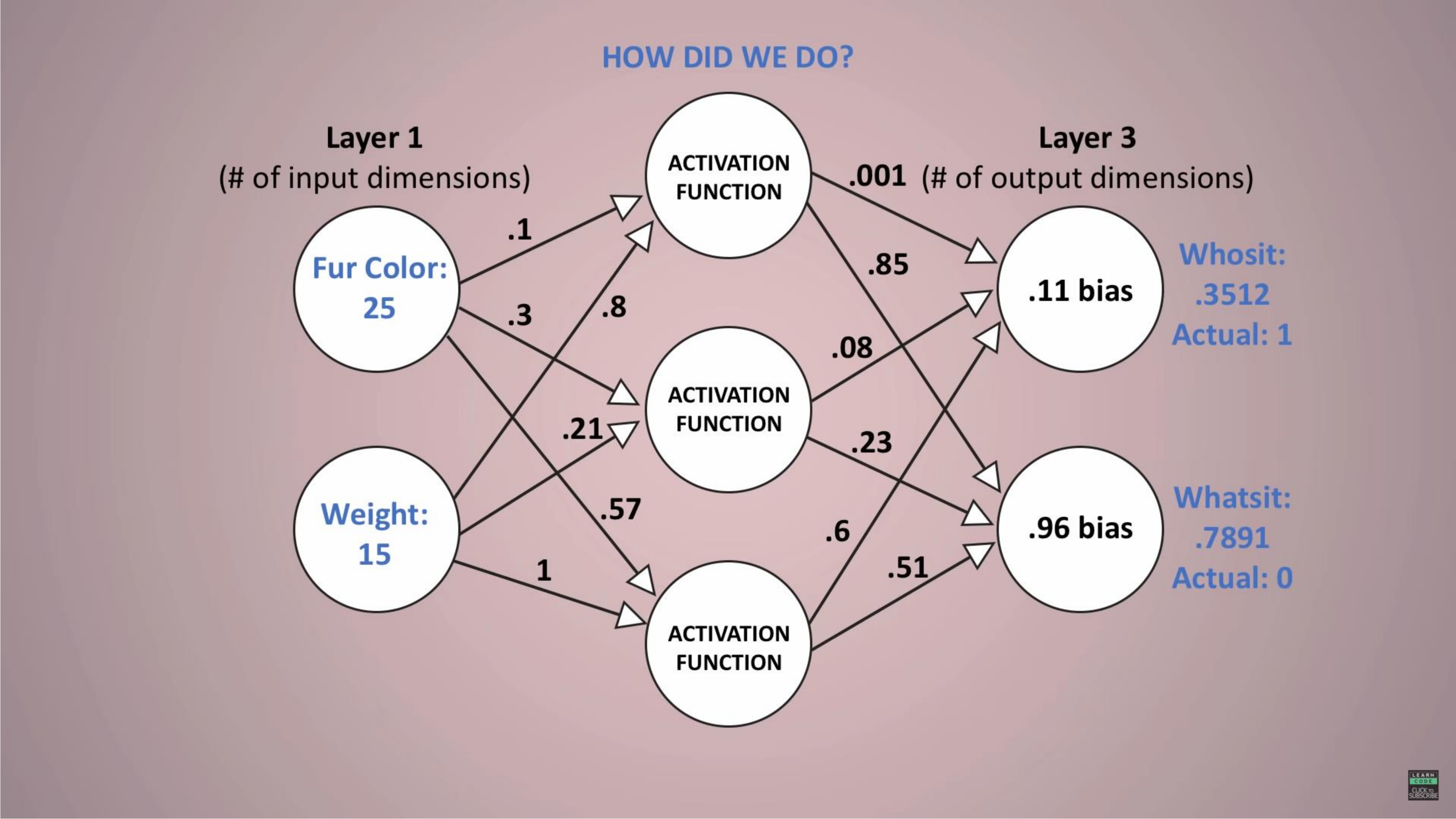


- **Wiederholung**
- **Aufbau Neuronaler Netze (NN)**
- **Hyperparameter in NN**
- **Frameworks zur Implementierung von NN**
- **Implementierung eines NN mit TensorFlow und Python**
- **Datenaufbereitung für TensorFlow**

# BEISPIELSKRIPT FÜR EINE SVM

**[Siehe `example_svm_estimation.zip`]**

# FRAGEN



# **HYPERPARAMATER IN NEURONALEN NETZEN**

- Anzahl der Hidden Layer des Netzes
- Anzahlen der Neuronen der Hidden Layer
- Wahl der Parameter der Optimierungsfunktion
- Wahl der Optimierungsfunktion
- Wahl der Kostenfunktion
- Aktivierungsfunktionen der Hidden Layer

# PARAMETER DES „ADAM“ OPTIMIZERS

- **alpha (learning rate):**  
Lernparameter/Schrittweite der Optimierung
- **beta1 and beta2 (momentum):**  
Trägheit der Optimierung

# PYTORCH





# KERAS IN PYTHON UND R






Keras ist eine Schichtstelle / ein Funktionswrapper zur vereinfachenden Nutzung von TensorFlow

Zwei Nutzungsvarianten möglich:

- über das Package „keras“  
(vgl. <https://keras.rstudio.com/>)
- über die Nutzung von keras in Python und die Integration über das Paket „reticulate“.



# DATENAUFBEREITUNG FÜR TENSORFLOW

```
#####  
### Preparation of the Environment   
#####  
### Function Definition   
#####  
### Data Import   
#####  
### Data Preparation ###  
  
# Recoding of the variables into one-hot encoded (dummy) variables  
dummy_list <- c("view", "condition")  
house_pricing_dummy = dummy_cols(house_pricing, dummy_list)  
  
# Definition of lists for each one-hot encoded variable (just to make the handling easier)  
condition_dummies = c('condition_1', 'condition_2', 'condition_3', 'condition_4', 'condition_5')  
view_dummies = c('view_0', 'view_1', 'view_2', 'view_3', 'view_4')  
  
# Standardization of all variables (features and label)  
norm_list <- c("price", "sqft_lot", "bathrooms", "grade", "waterfront", view_dummies, condition_dummies) # list of all relevant variables  
norm_values_list <- get.norm_values(house_pricing_dummy, norm_list) # Calculation of the means and standard deviations  
house_pricing_norm <- norm_cols(house_pricing_dummy, norm_values_list) # Standardization of the variables  
  
#####  
### Selection of the Feature Variables and the Label Variable   
#####  
### Selection of Training and Validation data 
```

# AUFGABEN

- Trainiert bitte bis zur nächsten Woche ein neuronales Netz für Euren Datensatz und erstellt eine Vorhersage für den 05.06.2019
- Schaut [dieses Video](#) (5 Minuten) zu Zeitreihenanalysen an.