



Berner Fachhochschule  
Haute école spécialisée bernoise  
Bern University of Applied Sciences

# 2024 FS CAS PML - Supervised Learning

## 2 Klassifikation

### 2.4 Neuronale Netze

Werner Dähler 2024

## 2 Klassifikation - AGENDA

- 21. Instanzbasierte Modelle
- 22. Regelbasierte Modelle
- 23. Mathematische Modelle
- 24. **Neuronale Netze**
  - 241. **MLPClassifier**
- 25. Multiklass Klassifikation

## 2.4.1 Klassifikation - Neuronale Netze

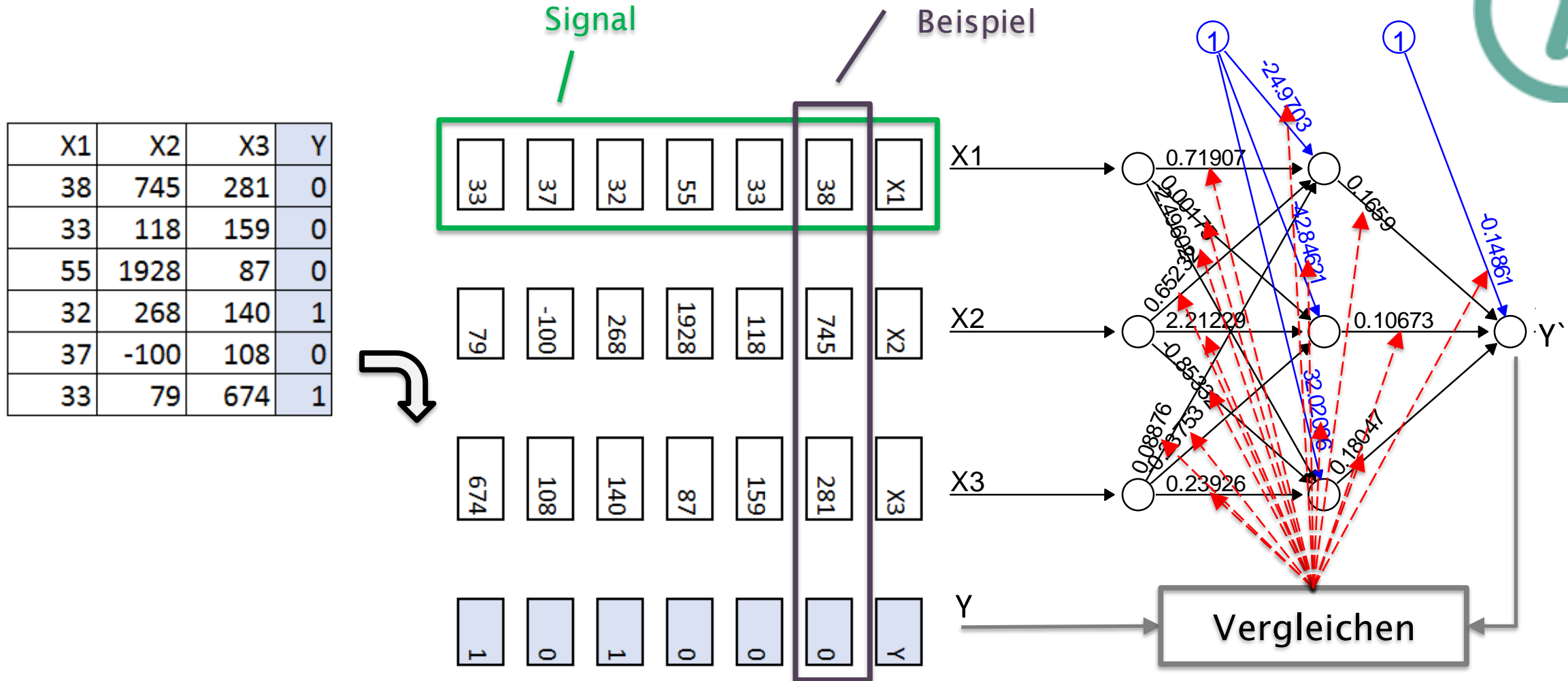
### 2.4.1.1 Theorie

- ▶ [MLPClassifier](#): Multi-layer Perceptron classifier (eine von drei Klassen des Moduls `sklearn.neural_network`)
- ▶ dieses Thema wird in einen separaten Modul (NeuroNet) unterrichtet, daher hier nur eine Kurzeinführung mit etwas Beispielcode für Scikit-learn



## 2.4.1 Klassifikation - Neuronale Netze

### 2.4.1.1 Theorie



## 2.4.1 Klassifikation - Neuronale Netze

### 2.4.1.1 Theorie

- ▶ die Features ( $X_1$ ,  $X_2$ ,  $X_3$ ) werden als Eingangssignale an die Input Ports des Netzes übergeben
- ▶ die Signale werden von Knoten zu Knoten weitergereicht, wobei sie auf den Verbindungen verstärkt oder gedämpft werden (Gewichtungsfaktor  $w_i$ )
- ▶ in den Knoten werden die eingehenden Signale addiert, mittels einer Aktivierungsfunktion (meist Logistische Regression, vgl. Kap. 2.3.4) umgerechnet und am Ausgang für die Weitergabe an die nächsten Knoten angeboten
- ▶ das Ausgangssignal des Netzes ( $y'$ ) wird mit dem wahren Wert von  $y$  verglichen und die Verstärkungsfaktoren auf den Verbindungen iterativ angepasst, bis die Differenz zwischen  $y$  und  $y'$  minimal wird
- ▶ das "Gelernte" im Netz sind somit nicht die Knoten selber, sondern die ermittelten Gewichte auf den Verbindungen zwischen den Knoten



## 2.4.1 Klassifikation - Neuronale Netze

### 2.4.1.2 Praxis

```
from sklearn.neural_network import MLPClassifier
model = MLPClassifier(random_state = 1234)
model.fit(X_train, y_train)
print(model.score(X_test, y_test))
```

0.7681620839363241

- ▶ die Performance ist im Vergleich mit den anderen bisher behandelten Klassifikatoren relativ schlecht
- ▶ ausserdem recht **instabil**, d.h. wiederholtes Trainieren und Testen mit unterschiedlichen Werten von `random_state` führt zu einer verhältnismässig grossen Streuung der ermittelten Performance (vgl. rechts und [ipynb])

