

Ausarbeitung Versuch 3 ILS Jan Holderied und Martin Goien

Aufgabe 1

a)

- `softmax(a)`: Berechnet die Softmax funktion für eine potentiellen Vektor `a`. Die Softmaxfunktion ist die Normalisierte Exponentialfunktion. `a` ist ein Vektor der Dentratischen Potentials. Returned einen Vektor der gleichen länge wie `a`.
- `def forwardPropagateActivity(x,W1,W2,flagBiasUnit=1)`: Propagiert die Neuronale Aktivität durch das Netzwerk in die forwärts Richtung. Startet beim Input und Endet beim Output.
- `def backPropagateErrors(z_1,z_2,t,W1,W2,flagBiasUnit=1)`: Zurückpropagieren der Fehlersignale die rückwärts durch das Netzwerk. Bedeutet von der Outputlayer zur Inputlayer.
- `def doLearningStep(W1,W2,xn,tn,eta,lmbda_by_N=0,flagBiasUnit=1)`: Einen Lernschritt vornehmen mit Input Datenvektor und dem dazugehörigen Zielvektor. Innerhalb des Lernschritts wird gleich der Backpropagation Algorithmus angewandt. Rückgabewert ist die geupdatete Gewichtsmatrix für jede Layer.
- `def getError(W1,W2,X,T,lmbda=0,flagBiasUnit=1)`: Berechnet den Kreuzentropiefehler für das gesamte Datenset für die MLP Gewichtsmatrizen `W1` und `W2`. Zurückgegeben wird der Final berechnete Fehler.
- `plotDecisionSurface(W1,W2,gridX,gridY,dataX1,dataX2,contlevels,epoch,flagBiasUnit=1)`: Zeigt als Plot die Klassengrenzen Oberfläche nach Training des Modells.

Aufgabe 2

Leider haben wir keine Vergleichswerte zu Versuch Nummer eins, da das Kernel MLP Modell dort nicht richtig funktioniert hat.

MLP Klassifikator

Fold 1: Training Accuracy - 1.0, Test Accuracy - 0.975 Fold 2: Training Accuracy - 1.0, Test Accuracy - 0.85 Fold 3: Training Accuracy - 1.0, Test Accuracy - 0.95 Fold 4: Training Accuracy - 1.0, Test Accuracy - 0.9743589743589743 Fold 5: Training Accuracy - 1.0, Test Accuracy - 1.0

KNN Klassifikator

Fold 1: Training Accuracy - 0.9696969696969697, Test Accuracy - 0.8787878787878788 Fold 2: Training Accuracy - 0.9621212121212122, Test Accuracy - 0.9545454545454546 Fold 3: Training Accuracy - 0.946969696969697, Test Accuracy - 0.9696969696969697

Aufgabe 3

MLP Regressior

Das Regressionsmodell von Skicit Learn hat leicht schlechtere Werte erzieht wie das Modell aus Versuch zwei.
Versuch Nummer zwei: MAE= 1.8183901930172532 MAPE= 0.014768012089478923
Versuch Nummer drei:

- Fold 1: Training MAE - 3.615663483795237, Test MAE - 3.6853852757904657
- Fold 2: Training MAE - 3.5777452232104596, Test MAE - 3.4788430446284795
- Fold 3: Training MAE - 3.403344319091153, Test MAE - 3.4949268117355277

KNN Regressionsmodell

Die Auswertung des KNN Regressor Modells hat ungefähr ähnliche Werte in der Evaluierung erzielt wie das KNN Modell aus Versuch Nummer zwei.

Versuch Nummer zwei: MAE= 2.0039946737683088 MAPE= 0.01617302367995536

Versuch Nummer drei: Training MAE - 1.5392083832335324, Training MAPE - 0.012518913724188592, Test MAE - 4.040360000000001, Test MAPE - 0.03238929595432739