Übungsblatt: MLP & Overfitting

Michel Bünger

NN.MLP.02: Vorwärtslauf im MLP

Dimensionen

$w^{[1]}$: (64, 25)

$w^{[2]}$: (32, 64)

$w^{[3]}$: (4, 32)

$b^{[1]}$: (64, 1)

$b^{[2]}$: (32, 1)

$b^{[3]}$: (4, 1)

Matrix Notationen

- Matrix Notation für Vektor $a^{[1]}$:

$$\begin{bmatrix} a_1^{[1]} \\ \vdots \\ a_{64}^{[1]} \end{bmatrix} = \text{relu}\left(\begin{bmatrix} w_{11}^{[1]} & \cdots & w_{1\,25}^{[1]} \\ \vdots & \ddots & \vdots \\ w_{64\,1}^{[1]} & \cdots & w_{64\,25}^{[1]} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_{25} \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ \vdots \\ b_{64}^{[1]} \end{bmatrix} \right)$$

$$a^{[1]} = \text{relu}\left(w^{[1]} \cdot x + b^{[1]} \right)$$

- Matrix Notation für Vektor $a^{[2]}$:

$$a^{[2]} = \text{relu}\left(\begin{bmatrix} w_{11}^{[2]} & \cdots & w_{64\,1}^{[2]} \\ \vdots & \ddots & \vdots \\ w_{32\,1}^{[2]} & \cdots & w_{32\,64}^{[2]} \end{bmatrix} \cdot \begin{bmatrix} a_1^{[1]} \\ \vdots \\ a_{64}^{[1]} \end{bmatrix} + \begin{bmatrix} b_1^{[2]} \\ \vdots \\ b_{32}^{[2]} \end{bmatrix} \right)$$

$$a^{[2]} = \text{relu}\left(w^{[2]} \cdot a^{[1]} + b^{[2]} \right)$$

- Matrix Notation für Vektor $a^{[3]} = \hat{y}$:

$$a^{[3]} = \text{relu}\left(\begin{bmatrix} w_{11}^{[3]} & \cdots & w_{1\,32}^{[3]} \\ \vdots & \ddots & \vdots \\ w_{4\,1}^{[3]} & \cdots & w_{4\,32}^{[3]} \end{bmatrix} \cdot \begin{bmatrix} a_1^{[2]} \\ \vdots \\ a_{32}^{[2]} \end{bmatrix} + \begin{bmatrix} b_1^{[3]} \\ \vdots \\ b_4^{[3]} \end{bmatrix} \right)$$

$$a^{[3]} = \text{relu}\left(w^{[3]} \cdot a^{[2]} + b^{[3]} \right)$$

Mögliche Nutzen

DA es vier Ausgaben gibt, handelt es sich hier schonmal nicht um ein Binäres Problem handelt. Stattdessen könnte es sich hier um eine Multiklassen-Klassifikation oder eine one-hot-codierte Ausgabe handeln.

Außerdem, da in jedem Schritt relu verwendet wird, kann es sich hier auch nicht um eine Wahrscheinlichkeitsklassifikation handeln

Aufgrund dessen ist es sehr realistisch, dass es sich hier um eine mehrdimensionale Regression handelt.

NN.MLP.03: Tensorflow Playground

1. Logistisches Regressionsmodell

Gaussian Datensatz

Mit mindestens den Eingaben x_1 und x_2, erreicht die Entscheidungsgrenze einen Test- und Trainings loss von 0.000 sehr schnell, ohne sichtliche überanpassung.

Die Entscheidungsgrenze erreicht durchschnittlich nach ca. 125 Epochen den zuvor beschriebenen Zustand.

Aufgrund der sehr klar definierten Datsensätze im Gaussian Datensatz können bei allen Durchläufen, immer alle Punkte richtig klassifiziert werden.

Spiral Datensatz

Hier kann die Entscheidungsgrenze keine optimale plazierung finden, weshalb im durchschnitt nur ein Trainings-/ Test loss, von 0.5 erreicht wird. In manchen fällen, abhängig von der Testdaten auswahl, kommt es zu overfitting, bevor sich dei Entscheidungsgrenze festigt, meißtens ist der Grund für die suboptimale Entscheidungsgrenze, jedoch die verteilung des Datnsatzes.

Aufgrund dieses Datensatzes wird immer ein Großteil der Punkte falsch Klassifiziert.

Circle Datensatz

Für den Circle Datensatz werden immer mindestens die Eingabedaten x_1^2 und x_2^2 benötigt, um die beste Entscheidungsgrenze zu finden.

Diese wiederum, bildet sehr schnell eine annähernt optimale Grenze, d.h. sie erreich fast sofort einen Test-/Trainings loss von ca. $\tilde{0}.005$, daraufhin dauert es extrem lange, bis sie sich einem loss, von 0.000 annähert.

Aufgrund dessen können theoretisch Punkte falsch klassifiziert werden, jedoch ist dies extrem unwahrscheinlich, weil die Entscheidungsgrenze immernoch fast optimal ist.

Exclusive or Datensatz

Für diesen Datensatz werden immer die EIngabe daten x_1x_2 für das beste Ergebniss benötigt. Die Entscheidungsgrenze bildet sich hier auch sehr schnell, jedoch kommt das gleiche Problem, wie zuvor aus, dass die loss Werte sich dem optimal sehr langsam annähern, jedoch spielt das hier aucch eine relativ kleine Rolle, weil diese Werte sehr niedrig sind.

Deshalb können auch nur mit sehr geringer wahrscheinlichkeit, Punkte dlasch klassifiziert werden.

2. MLP

[Hidden Layer = HL, Neuron = N]

HL = 1, N = 2

Spiral Die Entscheidungsgrenze bildet bei den Eingabewerten x_1 und x_2 bei jeder einstellung eine Linie, welche die Spirale mittig durchteilt, Die beiden Perzeptrone in der Hidden Layer zeigen hierbei zwei Linien, welche in die gleiche Richtung zeigen. und erreicht loss-Werte von durchschnittlich ca. 0.470, andere Eingabewerte erreichen loss-Werte in dem selben Bereich. Jedoch, bei nicht-linearen Eingabewerten, kommt es bei längeren Laufzeiten, zu wachsendem Overfitting.

- [**Circle**] Die Eingabewerte für ein optimales Ergebnis, sind hier x_1^2 und x_2^2, wo sich dei loss-Werte einer optimalen 0.000 annähern, die zwei Perzeptrone zeigen hierbei, zum einem den inneren Kreis und das andere, alles ausßerhalb des inneren Kreises. Overfitting kommt hier nicht vor.

HL = 1, N = 3

Spiral Ein Optimales Ergebnis kann mit den Eingabewerten $x_1, x_2, sin(x_1)$ und $sin(x_2)$, erreicht werden. Die Berechnungszeit ist jedoch sehr lange, bis sich ein akzeptierbares Ergebnis bildet, welches auf einen Test loss, von ca. 0.116 kommt. Die Perzeptrone bilden vertikale und horizontale Formen, welche ähnlich zu sinus/cosinus hügeln aussehen. Selbst bei der langen Berechnungszeit, kommt kein Overfittin vor.

Circle Die Eingabewerte für ein optimales Ergebnis, sind hier x_1^2 und x_2^2, wo sich dei loss-Werte einer optimalen 0.000 annähern, die Perzeptrone zeigen hierbei, entweder den inneren Kreis, oder alles ausßerhalb des inneren Kreises. Overfitting kommt hier nicht vor.

HL = 1, N = 5

Spiral Ein Optimales Ergebnis kann am schnellsten mit den Eingabewerten $x_1, x_2, sin(x_1)$ und $sin(x_2)$, erreicht werden. Die Berechnungszeit ist trotzdem sehr lange, jedoch ist das Ergebnis deutlich besser, als vorherige Durchläufe, welches auf einen Test loss, von ca. 0.020 kommt. Die Perzeptrone bilden größtenteils vertikale und horizontale Formen, welche ähnlich zu sinus/cosinus hügeln aussehen. Tanh hat hier zu dem schnellstem Ergebnis geführt, ReLU ha es nicht zu solch einem Gutem ergebnis, wie die anderen geschafft, wegen Problemen mit Overfitting. Sigmoid hat das Ergebnis erst nach ca. 3,000 Epochen erreicht.

Circle Die Eingabewerte für ein optimales Ergebnis, sind hier x_1^2 und x_2^2, wo sich dei loss-Werte einer optimalen 0.000 annähern. Alle verfahren erreichen hier ein optimales Ergebnis in sehr schneller Zeit, jedoch hat ReLU öfters Perzeptrone, welche eher, die Form einer Sanduhr annehmen, als die eines Kreises, was meißtens der fall ist.

HL = 2, N = 5

Spiral Ein Optimales Ergebnis kann am schnellsten mit den Eingabewerten $x_1, x_2, sin(x_1)$ und $sin(x_2)$, erreicht werden. ReLU schafft es zwar ein besseres Ergebniss, als in vorherigen durchläufen zu erreichen, jedoch bei längerem durchläufen, kommt es wiederholt zu Overfitting. Tanh und Sigmoid sind sich hier sehr ähnlich, in Bewertung und geschwindigkeit,

Außerdem weisen die Perzeptrone bei beiden in der zweiten HL eine erkennbare Spiralen-Struktur auf.

Circle Die Eingabewerte für ein optimales Ergebnis, sind hier wieder x_1^2 und x_2^2, wo sich dei loss-Werte einer optimalen 0.000 annähern. Sigmoid hängt in der Bildung des Ergebnisses, vergleichsweise ziemlich hinterher, weil einige der Perzeptrone in der zweiten HL unklar dargestellt sind und "beschlagen", könnte man sagen, aussehen, Außerdem hat ReLU erneut, öfters Perzeptrone, welche eher, die Form einer Sanduhr annehmen, als die eines Kreises, was meißtens der fall ist.

HL = 3, N = 7

Spiral Ein Optimales Ergebnis kann am schnellsten mit den Eingabewerten $x_1, x_2, sin(x_1)$ und $sin(x_2)$, erreicht werden. ReLU schafft es zwar ein besseres Ergebniss, als in vorherigen durchläufen zu erreichen, jedoch weisen keine Perzeptrone eine Spiralen-Striktur auf uns könnten eher mit Flecken auf dem Diagram verglichen werden. Tanh ist hier extrem viel schneller, als Sigmoid, jedoch erreichen beide ein Ergebnis im gleichem bereich und beide weisen perzeptrone mit einer erkennbare Spiralen-Struktur auf.

Circle Die Eingabewerte für ein optimales Ergebnis, sind hier wieder x_1^2 und x_2^2, wo sich dei loss-Werte einer optimalen 0.000 annähern. Sigmoid hängt in der Bildung des Ergebnisses, vergleichsweise ziemlich hinterher, weil einige der Perzeptrone in der zweiten HL unklar dargestellt sind und "beschlagen", könnte man sagen, aussehen, Außerdem hat ReLU erneut, öfters Perzeptrone, welche eher, die Form einer Sanduhr annehmen, als die eines Kreises, was meißtens der fall ist.

HL = 4, N = 7

Spiral Hier ist zu bemerken, das für ReLU und Tanh mit erhöhten HL und Neuronen, beide Modelle instabieler werden und öfters Overfitting vorkommt, Sigmoid jedoch, wird immer langsamer, aber dafür ist das Ergbnis sehr stabil und es kommt nie Overfitting vor. Trotsdem hat Tanh hier in beinahe optimales ergebnis erreicht.

Circle Hier hat sich nicht im vergelich zu den vorherigen Durchführungen geändert, bis auf das verlangsamen der Sigmoid Durchführung, weil die

hinteren HL alle noch "beschlagen" sind, bis sich ein ergebnis bilden kann. Abseits dvon, ist aber alles gleich.