Deep Learning

2023-11-23

Ämne: FFNN for regression and classification, architecture, Activation functions, loss functions

Smartr

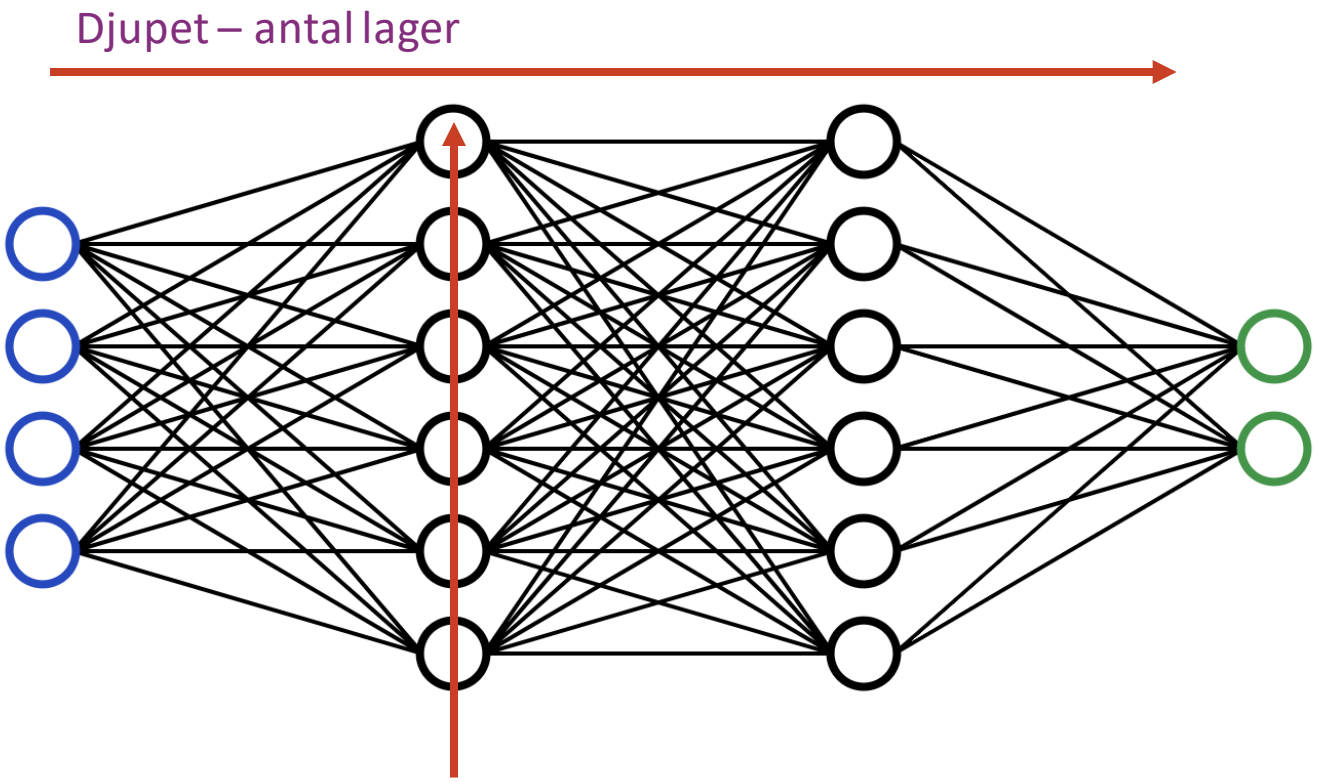
Agenda

- Arkitektur för neurala nätverk
- Aktiveringsfunktioner för regression och klassificering
- Loss funktioner för regression och klassificering

Neurala nätverk arkitektur

- Arkitektur på ett NN beskriver den övergripande strukturen – vilka beståndsdelar som finns och hur de sitter ihop.

- Arkitekturen på nätverket har stor betydelse för dess prestanda.



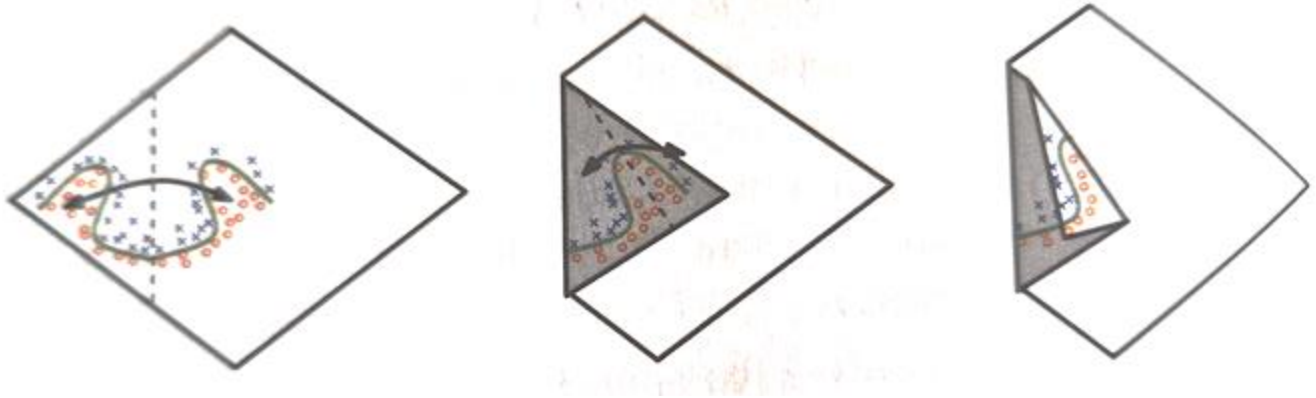
Smartr

Neurala nätverk arkitektur

- Hur påverkar bredden ett nätverk?
  - Input: Antal inputdimensioner
  - Output: Antal outputdimensioner
  - Hidden lager: Hur mycket ”beräkningsyta” eller ”minne” nätverket har att arbeta med.

Neurala nätverk arkitektur

- Hur påverkar djupet ett nätverk?
  - Djupare nätverk ger möjlighet till högre komplexitet och abstraktionsnivå i ett nätverk.



Bilden visar en intuitiv geometrisk representation av hur djupet bidrar till att modellen kan lösa mer komplexa problem. Med varje hidden layer så kan vi ”vika” hyperplanet för att hitta speglingar i funktionen, och på så sätt lösa en ”enklare” funktion.

Neurala nätverk

- För små/enkla nätverk kommer inte kunna lösa problemet.
- För stora/komplexa nätverk kommer inte kunna generalisera bra (de lärt sig datan utantill - overfitting) och kommer ta lång tid att träna.

Hur vet jag vilken arkitektur jag ska använda?

- Million dollar question!
- (förutom input och output – det vet vi)

- Det finns inga garantier för att ett nätverk lär sig.
- Titta på etablerade arkitekturer och testa.

Neurala nätverk input

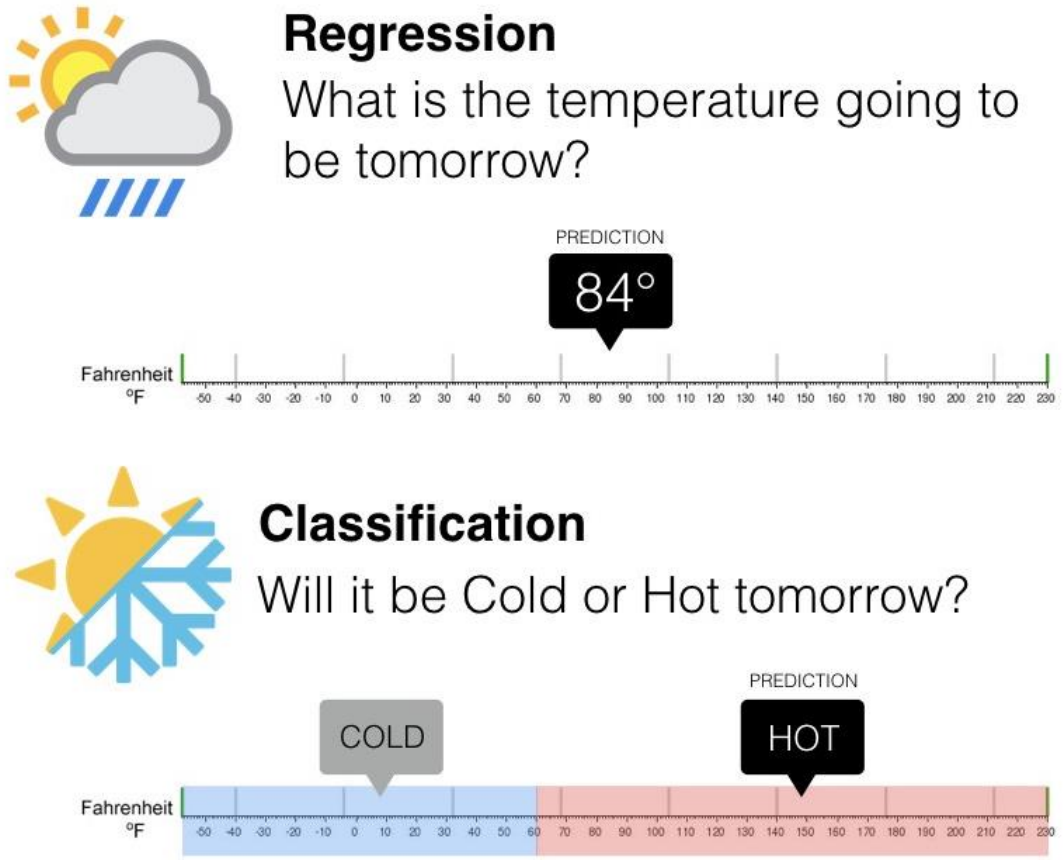
- Hur ska ditt input-lager se ut beroende på vad du har för data?

- MNIST – små bilder 28x28 pixlar.
- Koordinater – x,y,z
- Huspris

Neurala nätverk output

- Regression vs Klassifiering
- På outputen bör använda en aktiveringsfunktion också. Den skiljer sig ofta från aktiveringsfunktionerna i resten av nätverket och beror på vilket problem man försöker lösa.

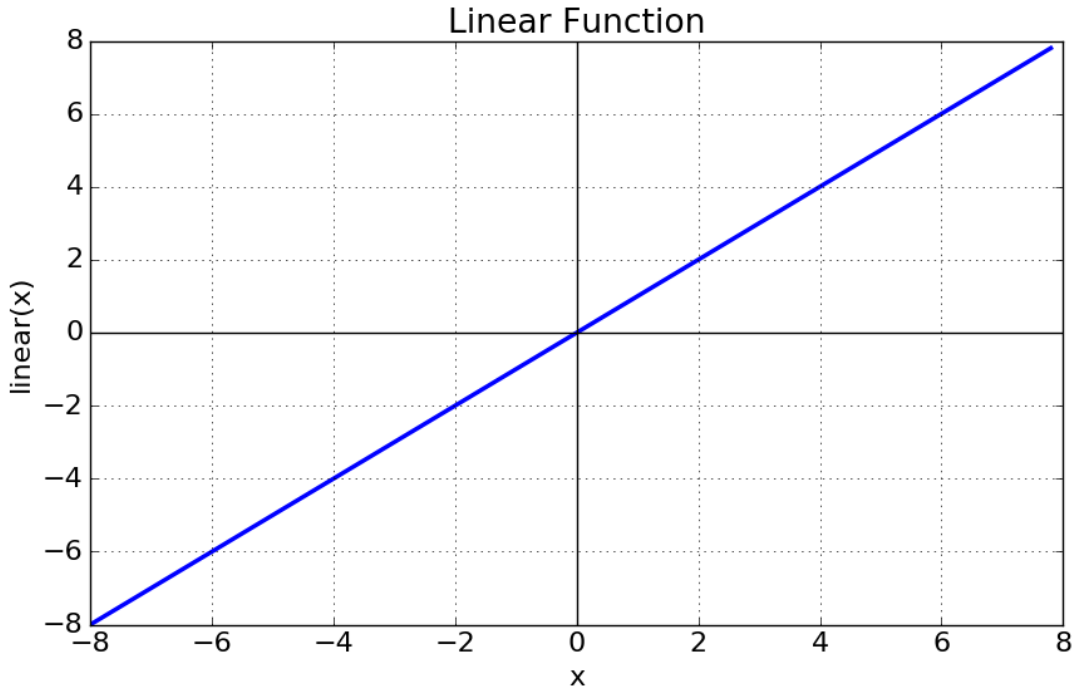
- Output activation functions:
  - R: Linear (because unbounded)
  - C: Simple Sigmoid or Softmax.



Neurala nätverk output aktiveringsfunktion - Regression

- Linjär aktiveringsfunktion är vanligast
- f(x) = x

- Behåller outputen som den är från nätverket, vilket passar bra för vi vill (oftast) ha ett flyttal $(-\infty, \infty)$

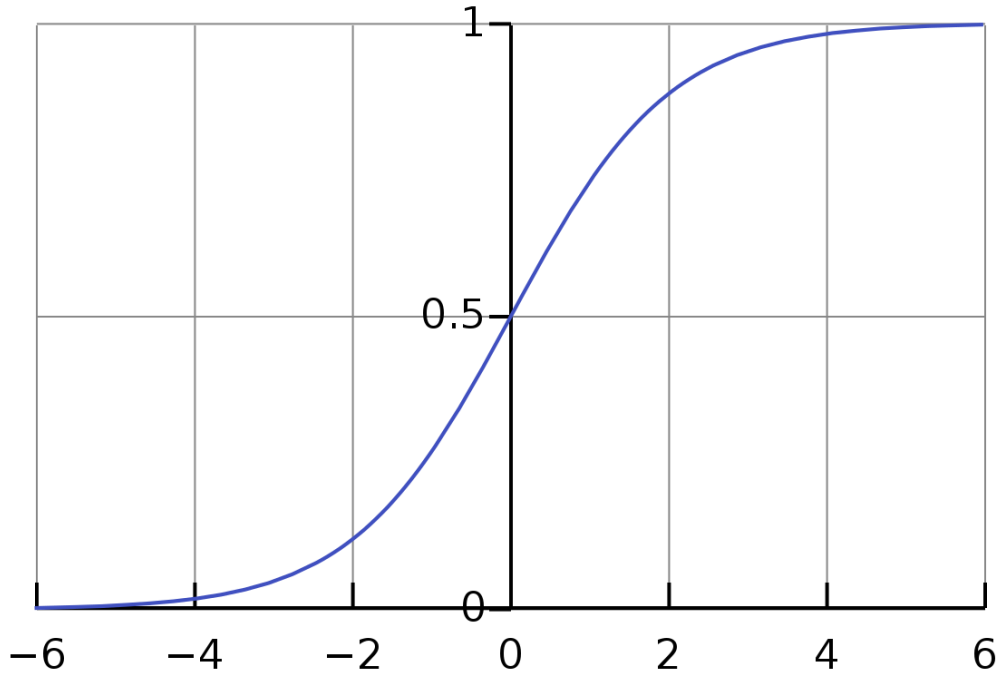


Binär klassifiering aktiveringsfunktion

- Sigmoid (logistic) funktion används då vi har binär klassifiering. Det gör om nätverkets output till en sannolikhet.

$$f(x) = \frac{1}{1 + e^{-x}}$$

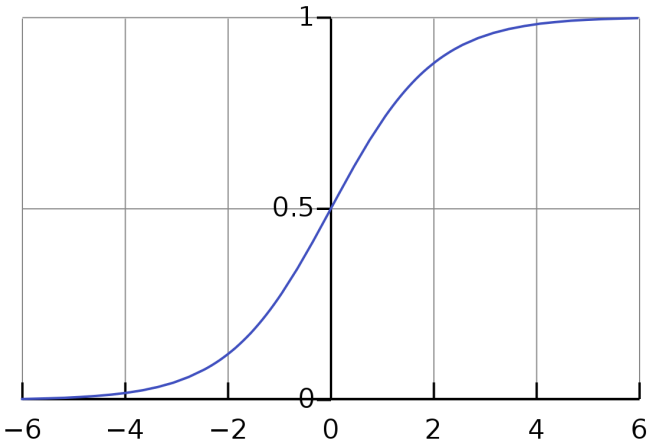
- Vid binär klassifiering har vi 1 output där vi svarar på frågan ”Vad är sannolikheten att inputdatat är med i klassen?”



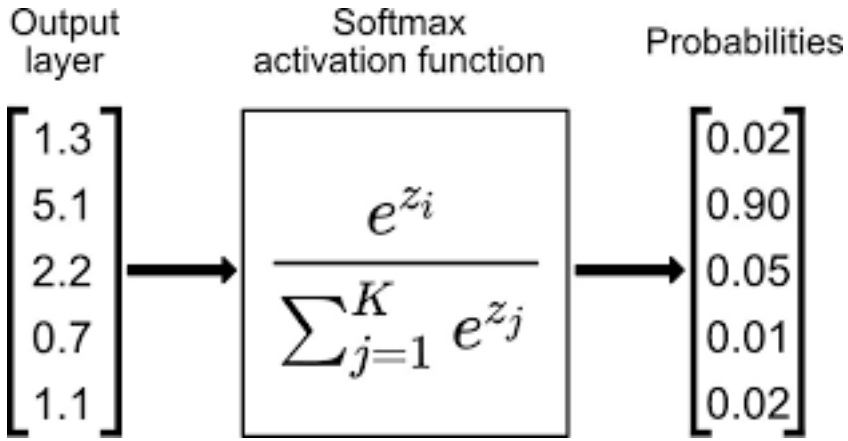
Multi-klass klassifiering aktiveringsfunktion

- Softmax används vid multi-klass klassifiering. Då har vi lika många output som klasser.

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K.$$



- Output blir en vektor med sannolikheten att inputdatat tillhör klass k.



Smartr

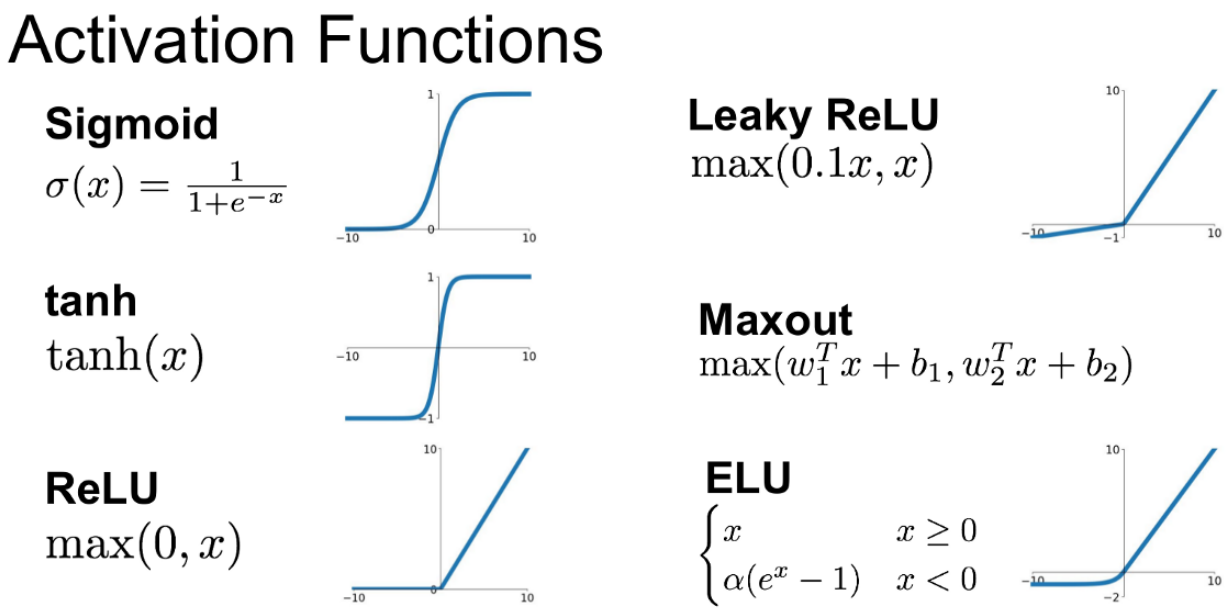
Aktiveringsfunktioner outputlagret

- Aktiveringsfunktion för output lagret:
  - Bestäms av vilket typ av problem vi vill lösa.
  - Nu kan ni 3 olika
    - regression, binär klassificering och multiklass klassificering.
  - Det finns många fler men dessa är vanligast för output lagret.

- För gömda lager används oftast andra aktiveringsfunktioner...

Aktiveringsfunktioner gömda lager

- Finns många att välja på.
- Vanligast är ReLU för den är ett enkelt sätt att introducera icke-linjäritet och extremt snabb att räkna på (derivatan är 1 eller 0).



Loss functions

- Ej att blanda ihop med aktiveringsfunktioner.
- Loss functions (eller cost function) används bara vid *träning* av nätverket.
- Ett mått på hur bra nätverket är jämfört med ground truth.
- Används sen för att beräkna uppdateringar för vikterna i nätverket.

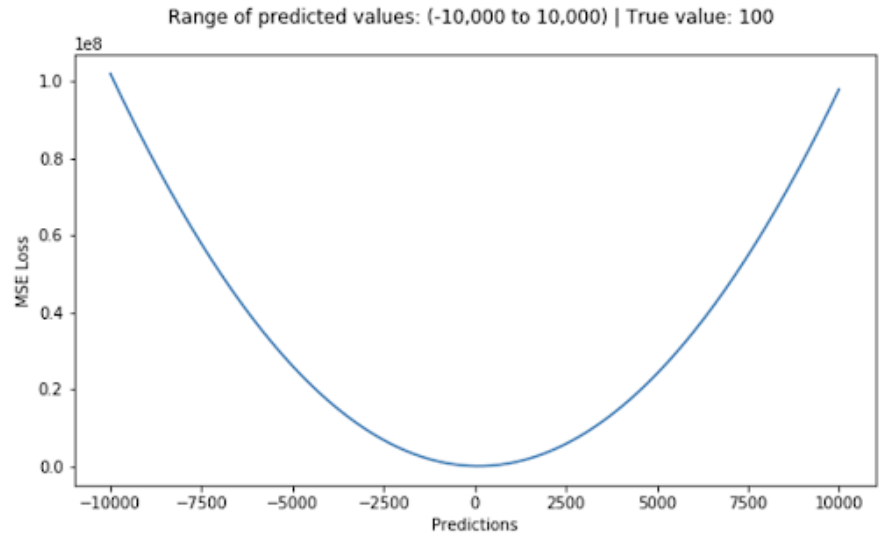
Note – In regression at the last neuron use linear activation function.

Loss function - MSE

Mean Squared Error (även kallad Squared loss eller L2 loss)

- Enklaste och vanligaste loss funktionerna för regression

$$\text{MSE} = \frac{1}{N}\sum_{i}^{N}(y_i - \hat{y}_i)^2$$



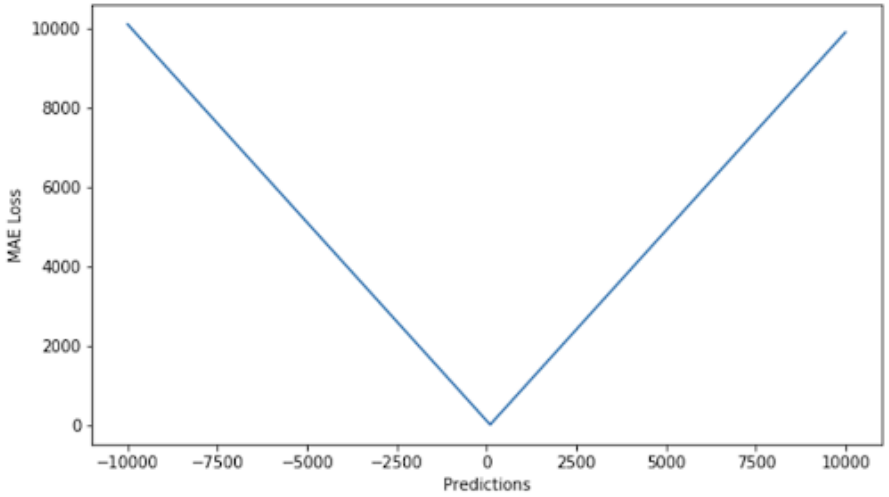
+ Lätt att förstå (avg. error i kvadrat)
+ Deriverbar (pga ^2)
+ Bara 1 lokalt minima
- Hanterar outliers dåligt eftersom stora fel blir ^2.

Note – In regression at the last neuron use linear activation function.

Loss function - MAE

- Mean absolute error (även kallad L1 loss)
- Också en av de enklaste och vanligaste loss för regression.

$$\text{MAE} = \frac{1}{N}\sum_{i}^{N}|y_i - \hat{y}_i|$$



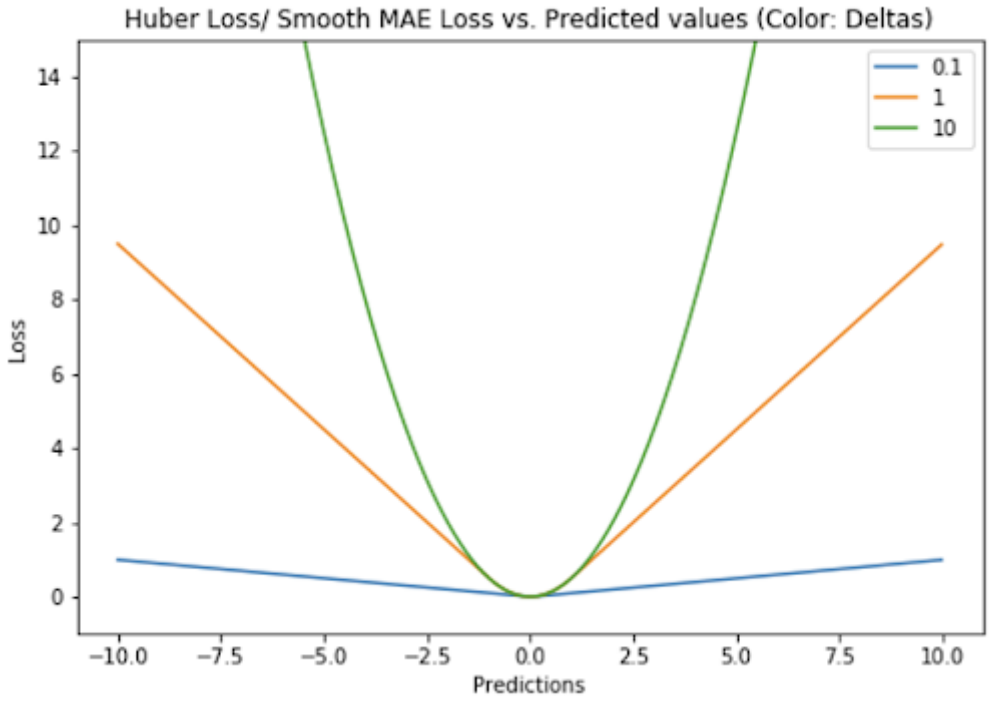
+ Lätt att förstå (genomsnitt error)
+ Mer robust mot outliers
- Ej deriverbar vid 0

Huber loss

- Mellanting mellan MSE och MAE. Kallas även Smooth MAE

$$Huber = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(y_i - \hat{y}_i)^2 \qquad |y_i - \hat{y}_i| \leq \delta$$

$$Huber = \frac{1}{n}\sum_{i=1}^{n}\delta\left(|y_i - \hat{y}_i| - \frac{1}{2}\delta\right) \qquad |y_i - \hat{y}_i| > \delta$$



+ Derivata vid 0
+ Bra på att hantera outliers
- Ökad komplexitet och kräver hyperparameteroptimering av δ

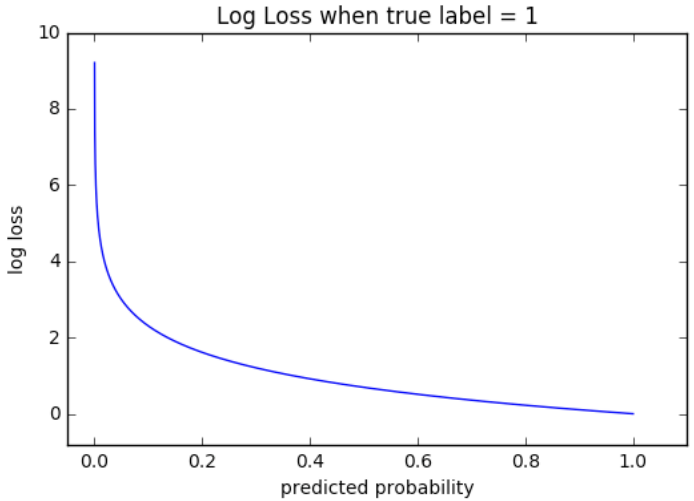
Note – In classification at last neuron use sigmoid activation function.

Binary Cross Entropy Loss

- Vanligaste loss funktionen för binär klassificering (även log loss)
- Jämför skillnaden mellan sannolikhetsdistributioner.
- Ju större skillnad mellan $\hat{y}_i$ och y_i desto större loss-värde.

$$L = -\frac{1}{N}\sum_i^N y_i \log\big(p(y_i)\big) + (1 - y_i)\log(1 - p(y_i))$$

+ Deriverbar
- Flera lokala minima
- Ej så intuitiv att förstå



Smartr

Note – In multi-classification at last neuron use softmax activation function.

Multi-Class Cross Entropy Loss

- Mest använda loss funktion för multi-class klassificering
- Samma som binary cross entropy men ej begränsat till 2 klasser. (även Categorical cross entropy).

$$\text{CE} = -\frac{1}{N}\sum_i^N \sum_j^k \left[y_{ij} \log\left(\hat{y}_{ij}\right)\right] = -\log\left(\frac{e^{s_j}}{\sum_j^c e^{s_j}}\right)$$

\+ Snabb att räkna på

\+ Konvergerar ofta relativt snabbt

\- Känslig mot outliers och obalanserad data.

Note – In multi-classification at last neuron use softmax activation function.

Kullback Liebler Divergence Loss

- Loss funktion för klassificering som baseras på KL Divergence
- Mäter hur mkt en sannolikhetsfördelning skiljer sig från en annan.

$$D\_KL(P \,||\, Q) = \sum P(x) * \log(P(x) / Q(x))$$

- KL Divergence Loss används med fördel vid unsupervised learning. Cross entropy är bättre vid supervised learning där det finns ground truth labels.

Träna nätverk - Överblick

Informationen flödar likt

1. Datapunkten x kommer in i nätverket via input noderna.
2. Informationen flödar genom nätverket genom hidden layers med aktiveringsfunktioner.
3. Vid output lagret så appliceras en output-aktiveringsfunktion.
4. Loss funktionen beräknar felet mellan outputen och ground truth.
5. Backprop + gradient descent för att ta reda på felet i varje vikt.
6. Uppdatera vikterna med ett litet steg.

Upprepa från steg 1 med nästa datapunkt.

Vikt initialisering

Det är standard att använda random initialisering för vikterna.

Varför funkar det inte att initialisera alla vikter till 0?

- Alla vikter i ett lager kommer då få samma derivata, samma viktuppdatering och ha samma värde.
- Vikt-bias blir helt meningslösa om de sätts till 0.

Inferens av nätverk - överblick

För att använda nätverket för inferens, eller forward pass, så flödar informationen likt:

1. Datapunkten k kommer in i nätverket via intput noderna.
2. Informationen flödar genom nätverket genom hidden layers med aktiveringsfunktioner.
3. Vid output lagret så appliceras en output-aktiveringsfunktion. Där har vi outputen från nätverket!