

Nätverk inom AI: Backprop

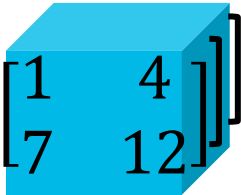
729G83

Översikt

- Allmän introduktion
- Forward pass (propagation of activation)
- Backward pass (propagation of error)
 - **Backpropagation of error el. backprop**
 - Beräkningsgrafer för att automatisera backprop
- Olika loss-funktioner
 - För klassificering
 - För regression

Allmän introduktion

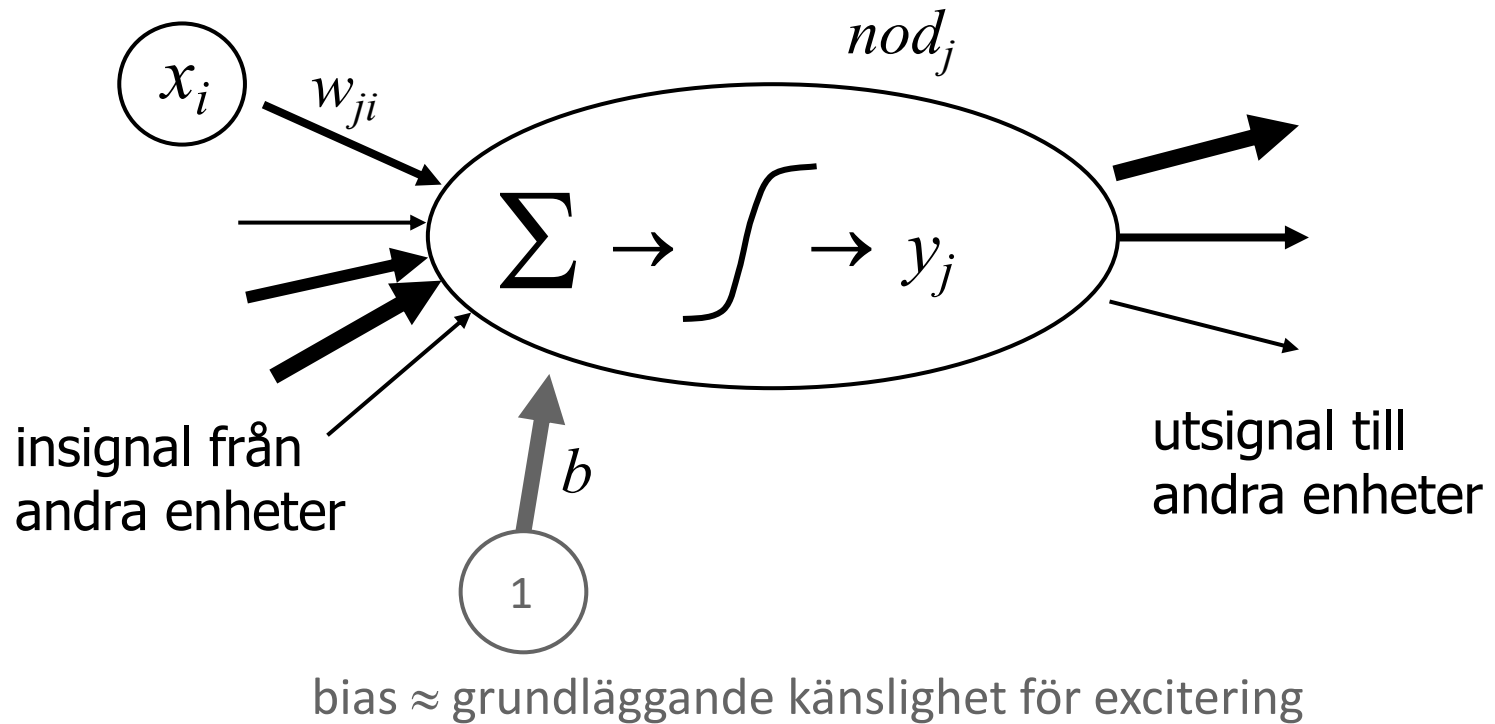
TensorFlow

- Vektor $\begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}$ shape = (3,)
- Matris $\begin{bmatrix} 2 & 4 & 1 \\ -1 & 6 & 8 \end{bmatrix}$ shape = (2, 3)
- Tensor  shape = (2, 2, 3)
(array)

Skillnaden mot emergent

- Kopplingar
 - Bara framåtkopplingar (feedforward)
- Inläring
 - Vi har inte två aktiveringar i varje nod som vi kan diffa mellan för att beräkna lokalt fel
 - *Felet* måste propageras bakåt

Aktiveringsfunktion

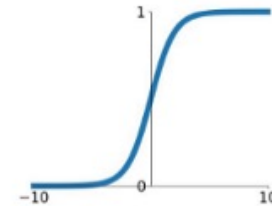


Olika aktiveringsfunktioner

- Ska vara
 - Kontinuerlig
 - Lätt att beräkna
- Mest populär: ReLU
 - Rectified Linear Unit

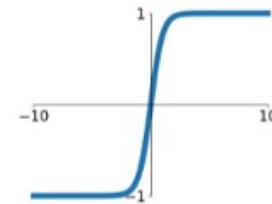
Sigmoid

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



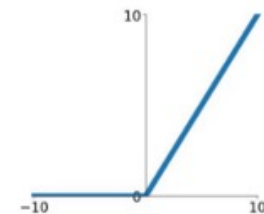
tanh

$$\tanh(x)$$



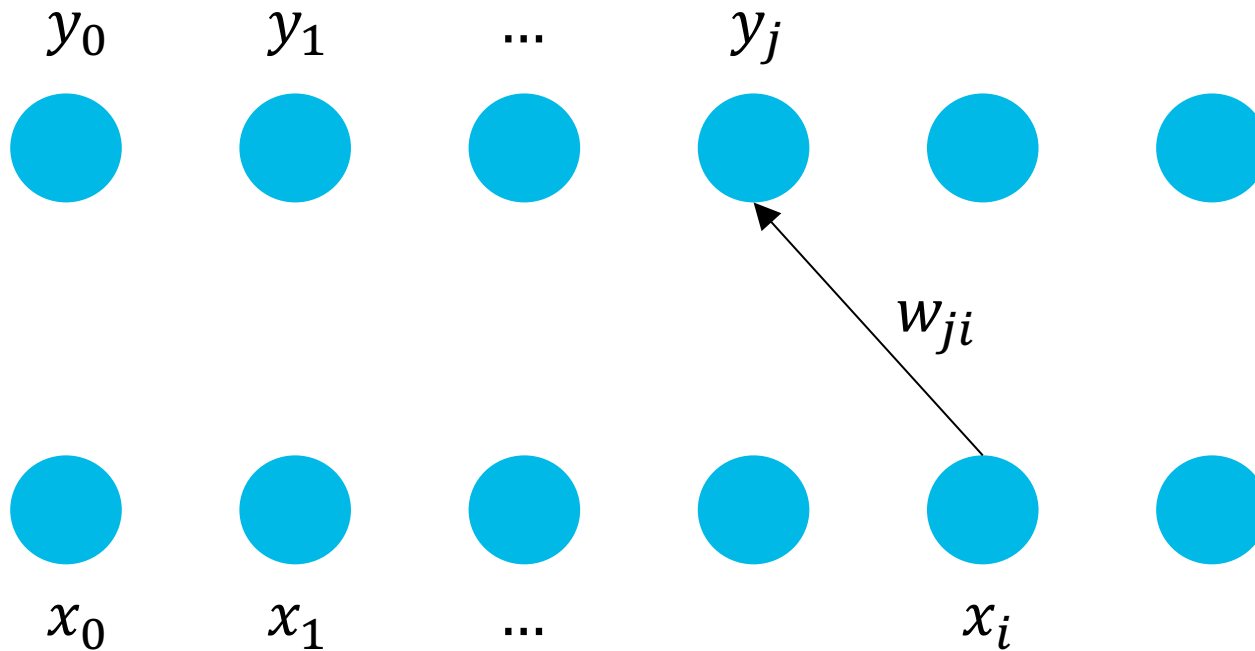
ReLU

$$\max(0, x)$$



Forward pass: propagering av aktiveringar

Lite notation

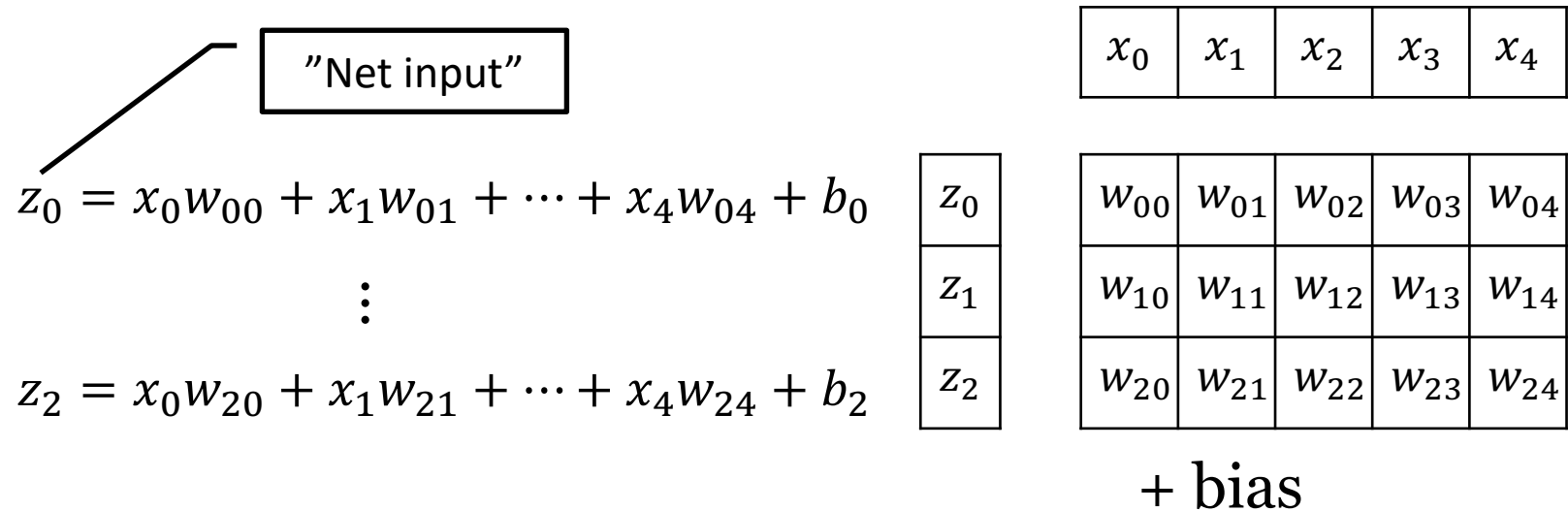


Viktmatris på rad j kolumn i

från

	x_0	x_1	...	x_i	
till					
y_0					
y_1					
...					
y_j				w_{ji}	

Först beräknas net input, sedan y



$$y_0 = \sigma(z_0) = \sigma(\mathbf{w}_0\mathbf{x} + \mathbf{b}_0)$$

$$\mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

Backpropagation of error: propagering av fel

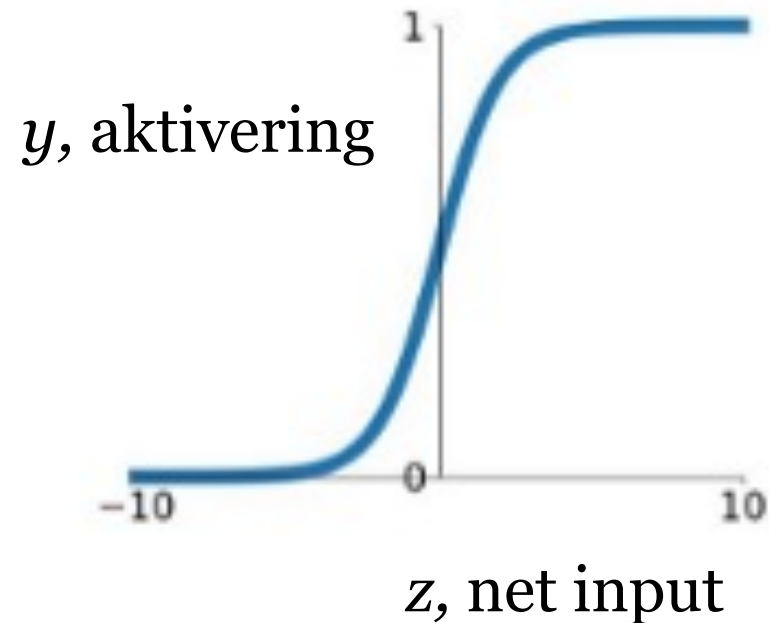
Propagering av fel

- Givet fel
 1. Beräkna hur felet skulle påverkas av en ändring av net input (summa inkommande signaler)
 - Beror på formen på aktiveringsfunktionen
 2. Hur net input i sin tur skulle förändras av ändring av inkommande vikter
 - Beror på viktens storlek och x_i

– Upprepa på föregående lager...
- När vi har riktningarna, uppdatera varje vikt åt rätt håll, så att felet minskas (lite grann)

1. Hur felet påverkas av net input (z)

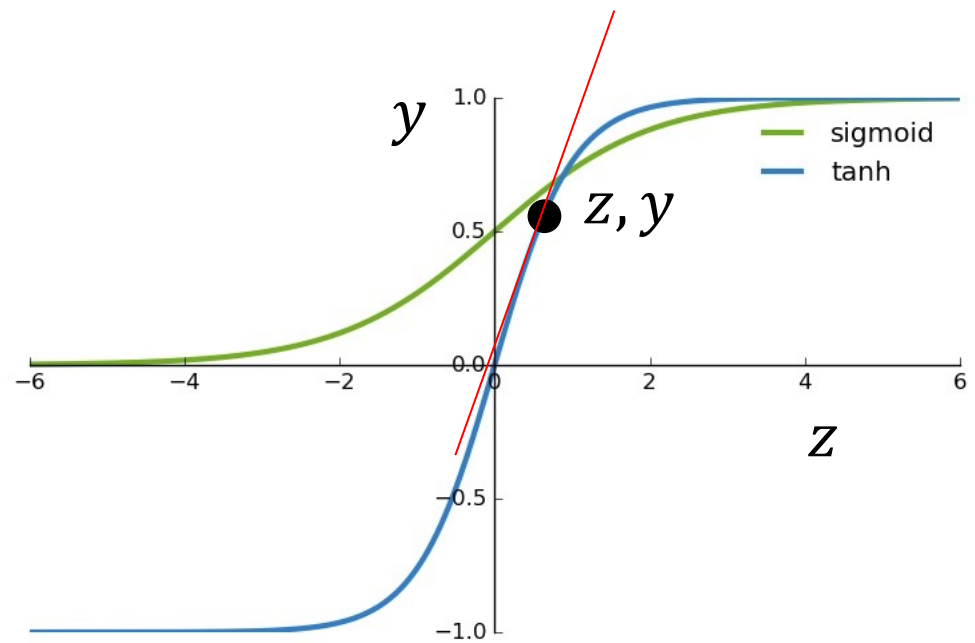
- Behöver veta hur funktionen mellan ut- och invariabel ser ut
- Hur skarpt förändras funktionen?
 - Brant funktion \rightarrow Felet hos y påverkas av små ändringar av net input



Derivata och gradient

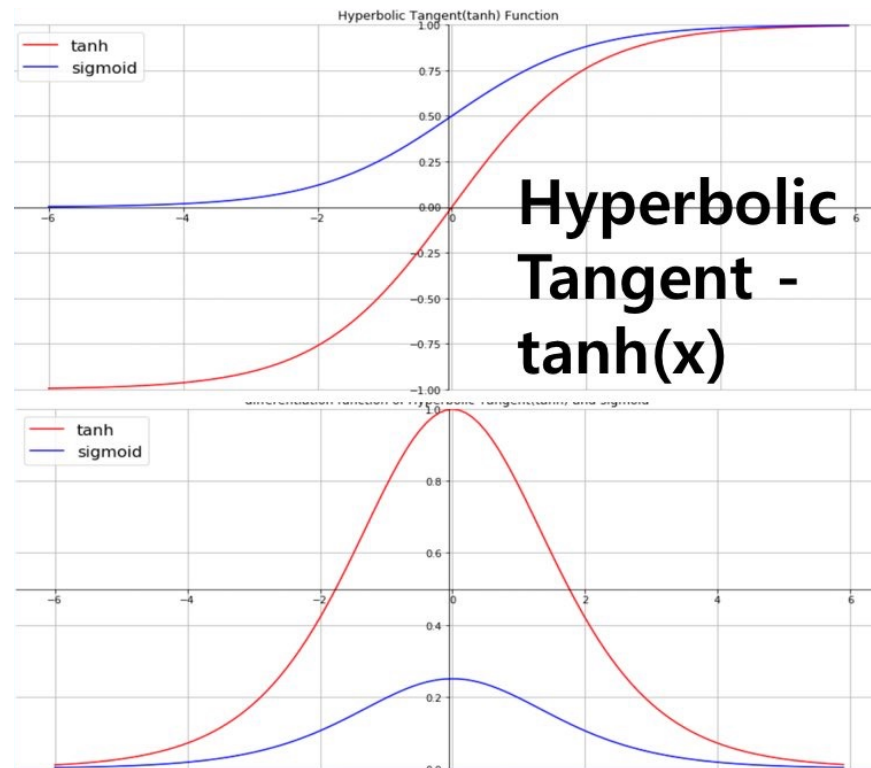
Fel och korrigering

- Om y ska minskas, hur ska z ändras?
 - Vill ha lutningen, i punkten z, y
- Derivatans av y
m.a.p. $z = \frac{dy}{dz} =$
 $\frac{\partial y}{\partial z} = y'$



Derivata = lutning i en viss punkt

- Hur mycket ändrar sig funktionsvärdet om vi pillar lite på *netin*
 - Brantast kring *netin* = 0
 - Kan ändra *y* väldigt effektivt kring *netin* = 0



Gradient

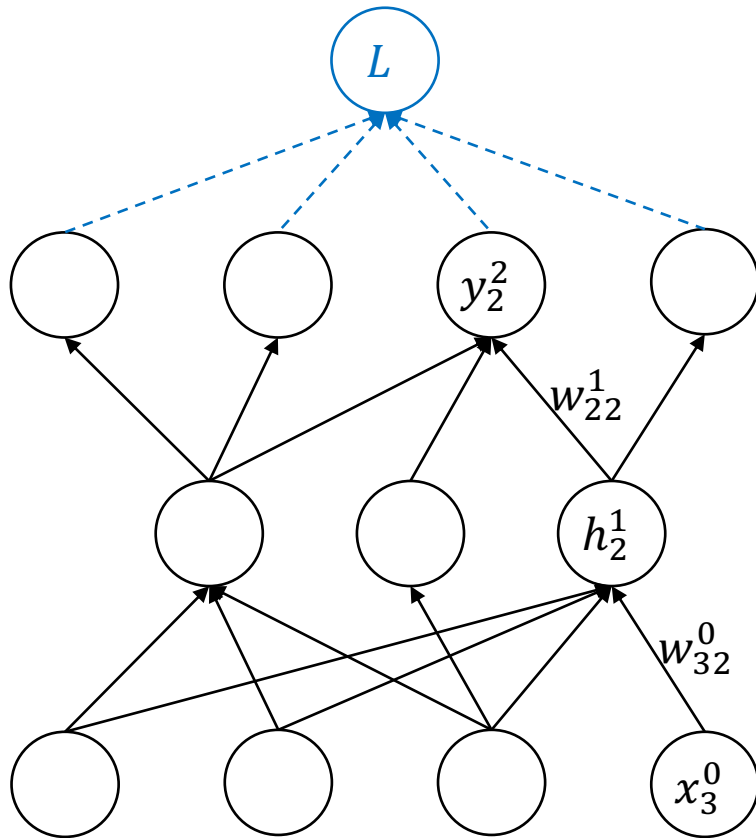
- Derivator över en vektor med flera element
 - T.ex. för alla vikter in till en nod
- Vektor med derivator i n riktningar i nuvarande punkt
 - Visar "landskapet" i en viss punkt
 - Hjälper till att bestämma i vilken riktning varje enskild invariabel ska ändras

Mini-batch

- Egentligen skulle man behöva köra igenom alla data, och sedan uppdatera vikterna
 - Så att vikterna funkar för alla data i data set:et
- Mini-batch = ett **slumpurval** från data set
 - T.ex. 32 stycken data (= samples) åt gången
 - Detta kallas ***Stochastic Gradient Descent***

Backprop

Forward pass



$$L = [(t_2^2 - y_2^2)^2 + \dots] + R, \text{ dvs. MSE} + \text{reg}$$

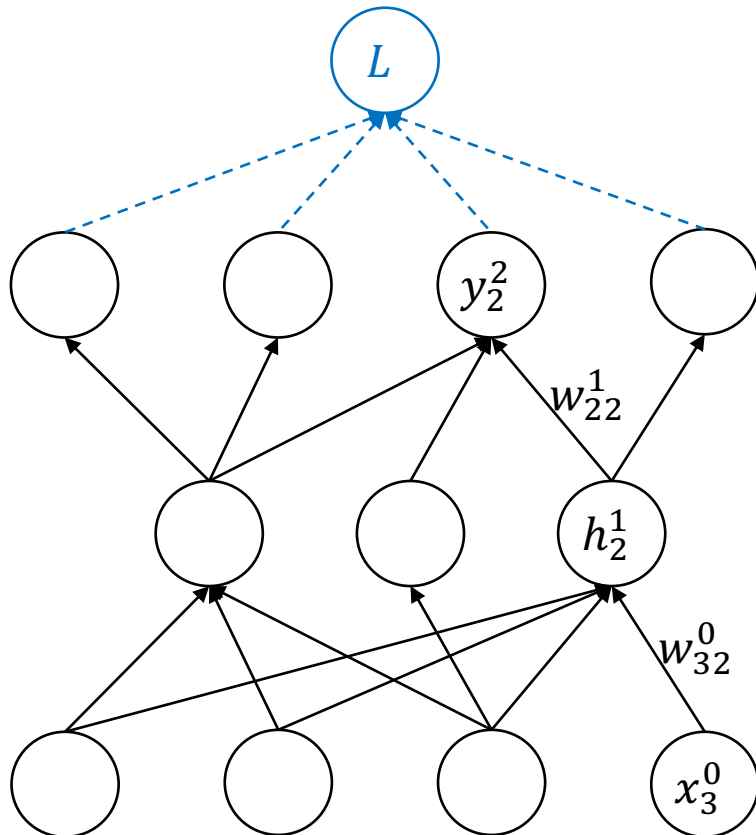
$$y_2^2 = \sigma(z_2^2)$$

$$z_2^2 = w_{22}^1 h_2^1 + \dots + b_2^2$$

$$h_2^1 = \sigma(z_2^1)$$

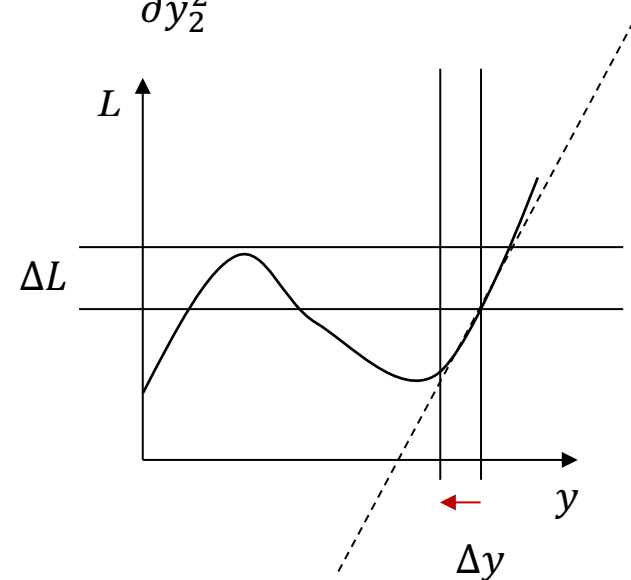
$$z_2^1 = w_{32}^0 x_3^0 + \dots + b_2^1$$

Bakåt-propagering av fel (intuition bakom)

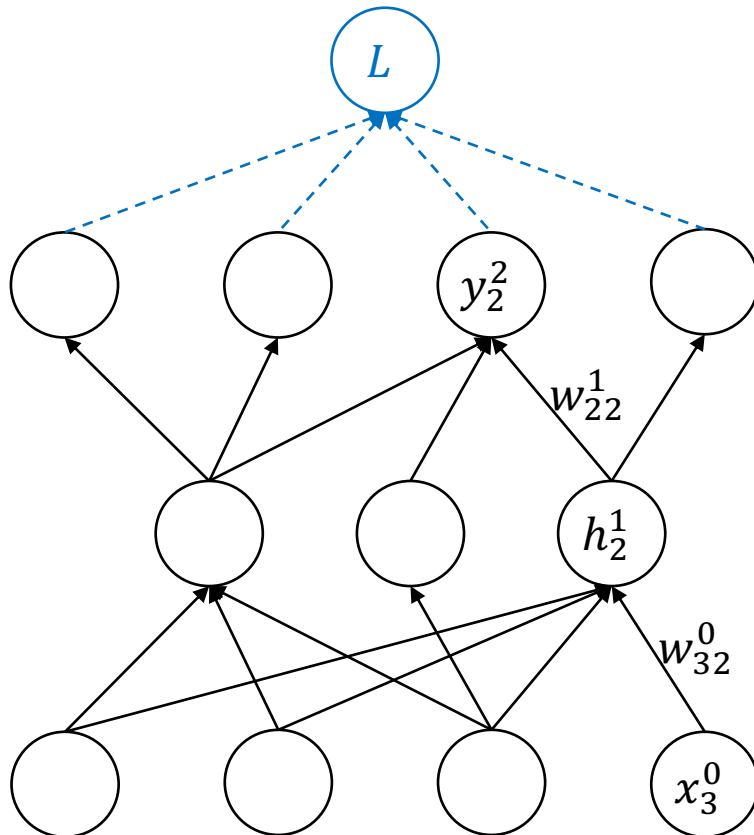


Hur ska y_2^2 ändras för att minska L ?

$$\Delta y_2^2 = -\eta \frac{\partial L}{\partial y_2^2}$$



Backward pass: propagering av "felet"

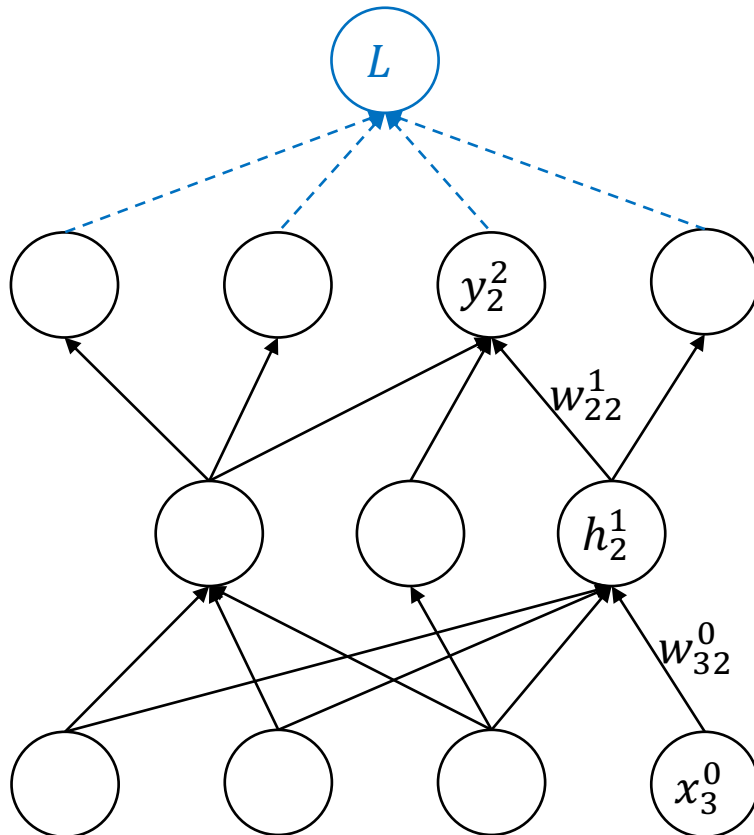


Hur ska y_2^2 ändras för att minska L ?
Dvs. hur påverkas L av y_2^2 ? Måste veta L'

Hur påverkas y av $w_{22}^1 = y'$

Hur påverkas h av $w_{32}^0 = h'$

Loss funktion av y , funktion av h , funktion...



$$L = \text{MSE}(y)$$

$$y = \sigma(wh + b)$$

$$h = \sigma(wx + b)$$

Men...

... vill veta hur w_{32}^0 ska ändras för att minska L
Dvs. hur påverkas L av w_{32}^0

Kedjeregeln för derivator

- För att få derivatan för loss m.a.p. vikter in till utlagret
 - Derivera loss m.a.p. y
 - Multiplicera med derivatan för y m.a.p. vikterna

$$(h(x))' = (g(f(x)))' = g'(f(x)) \cdot f'(x)$$

outside function

derivative of the inside

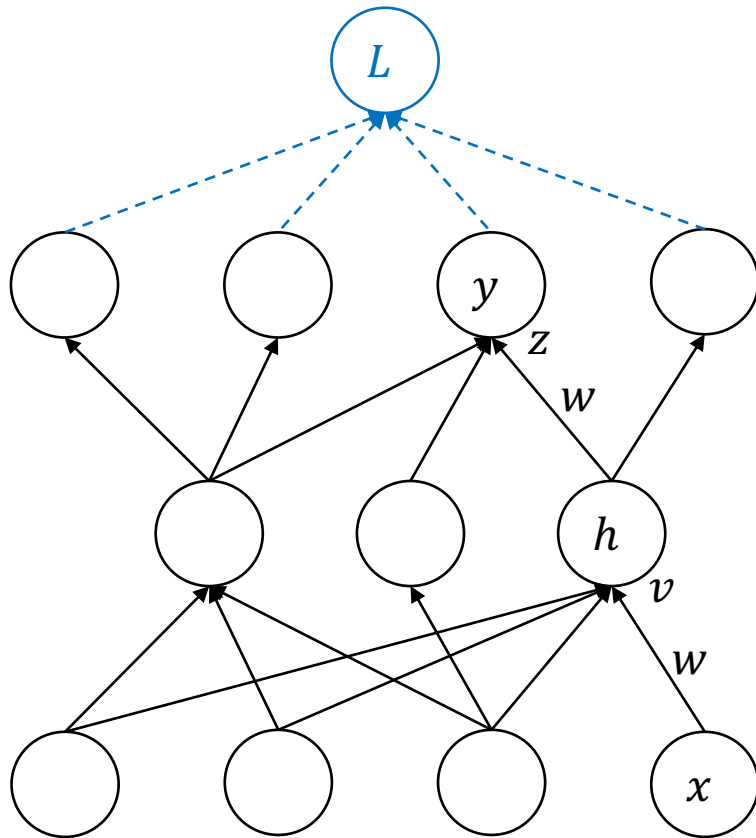
inside function

derivative of the outside at $f(x)$

Kedjeregeln

- Regeln säger att derivator ska multipliceras när man traverserar neråt mot input
 - Från topp-lagret, ner till lägsta lagret
 - Ju djupare nät, desto fler termer i multiplikationen

Multiplitera ihop lokala derivator



$$L = l(y)$$

$$y = \sigma(z)$$

$$z = wh + b$$

$$h = \sigma(v)$$

$$v = wx + b$$

$$l'(y)$$

$$\sigma'(z)$$

$$\frac{\partial z}{\partial h} = w, \frac{\partial z}{\partial w} = h$$

$$\sigma'(v)$$

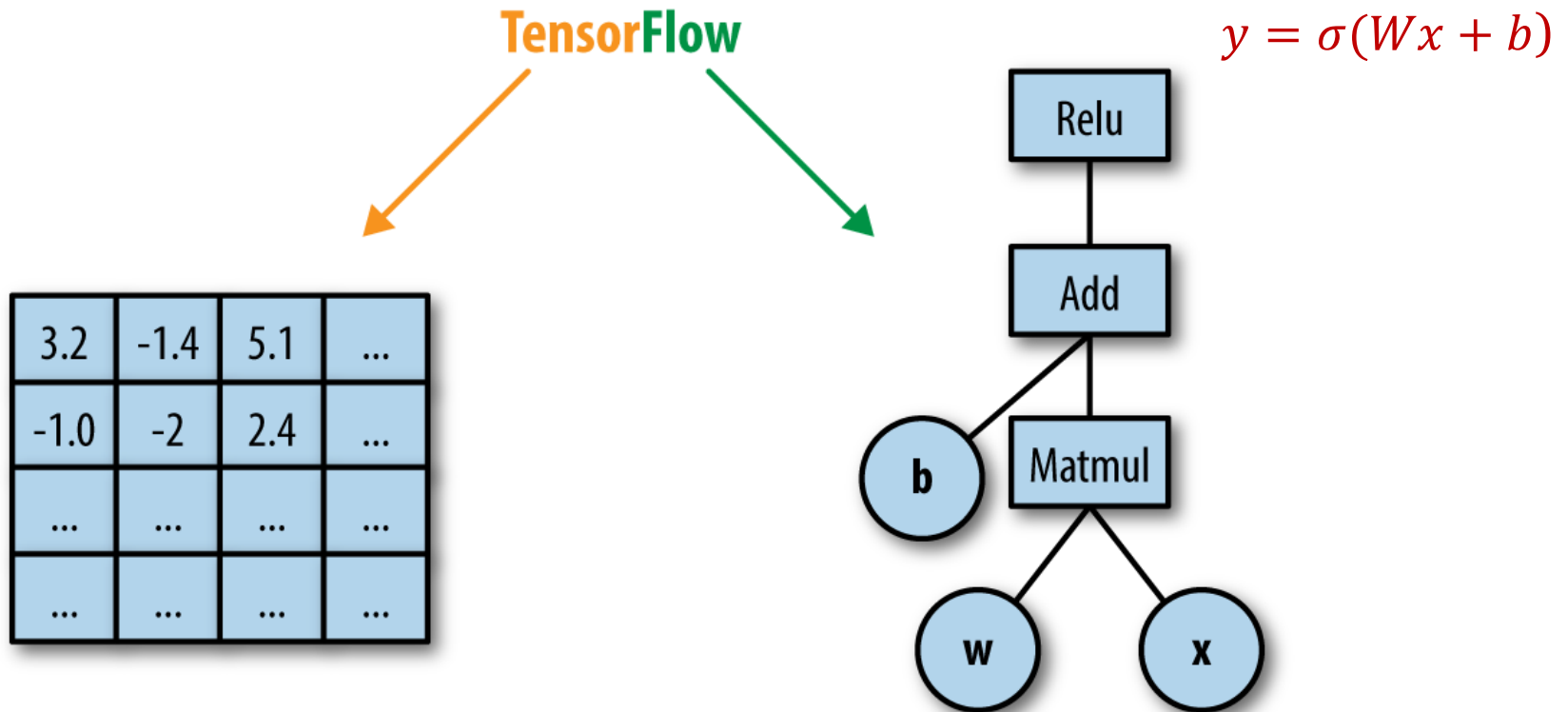
$$\frac{\partial v}{\partial x} = w, \frac{\partial v}{\partial w} = x$$

$$\frac{\partial L}{\partial w} = l'(y) \sigma'(z) w \sigma'(v) x$$

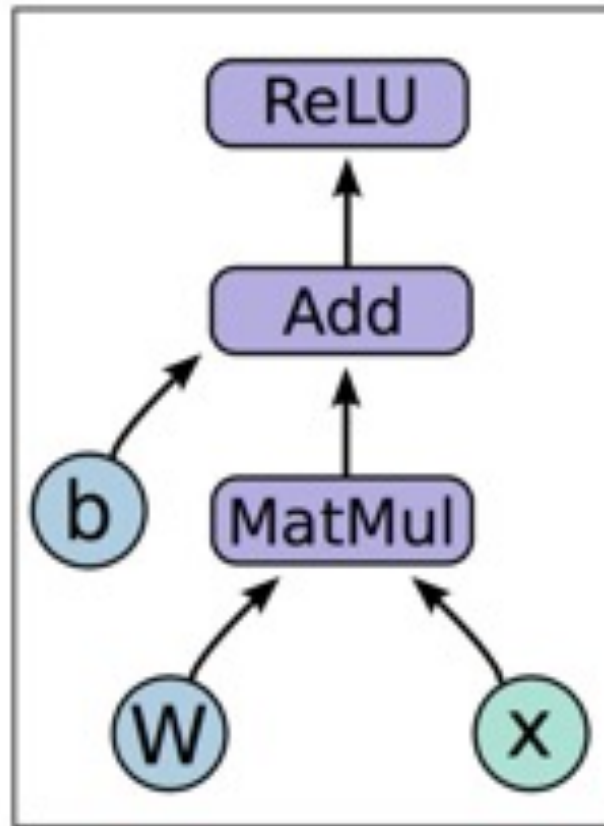
$$\Delta w = -\eta \frac{\partial L}{\partial w} = -\eta l'(y) \sigma'(z) w \sigma'(v) x$$

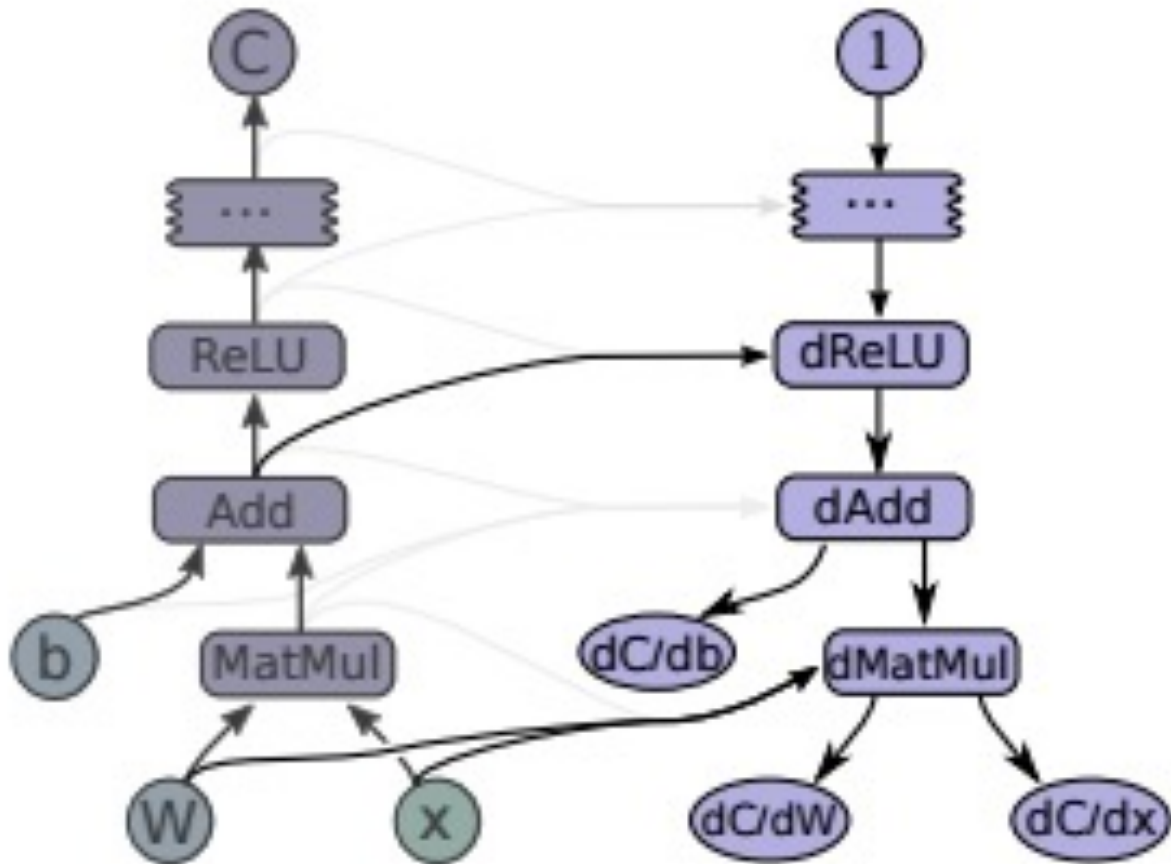
Beräkningsgraf (computational graph)

Aktiveringsflöde mellan två lager



$$h = \text{ReLU}(Wx + b)$$





Några derivator

1. $\frac{d}{dx}(c) = 0$, where c is a constant

2. $\frac{d}{dx}(x^n) = nx^{n-1}$, where n is any real number

3. $\frac{d}{dx}(e^x) = e^x$

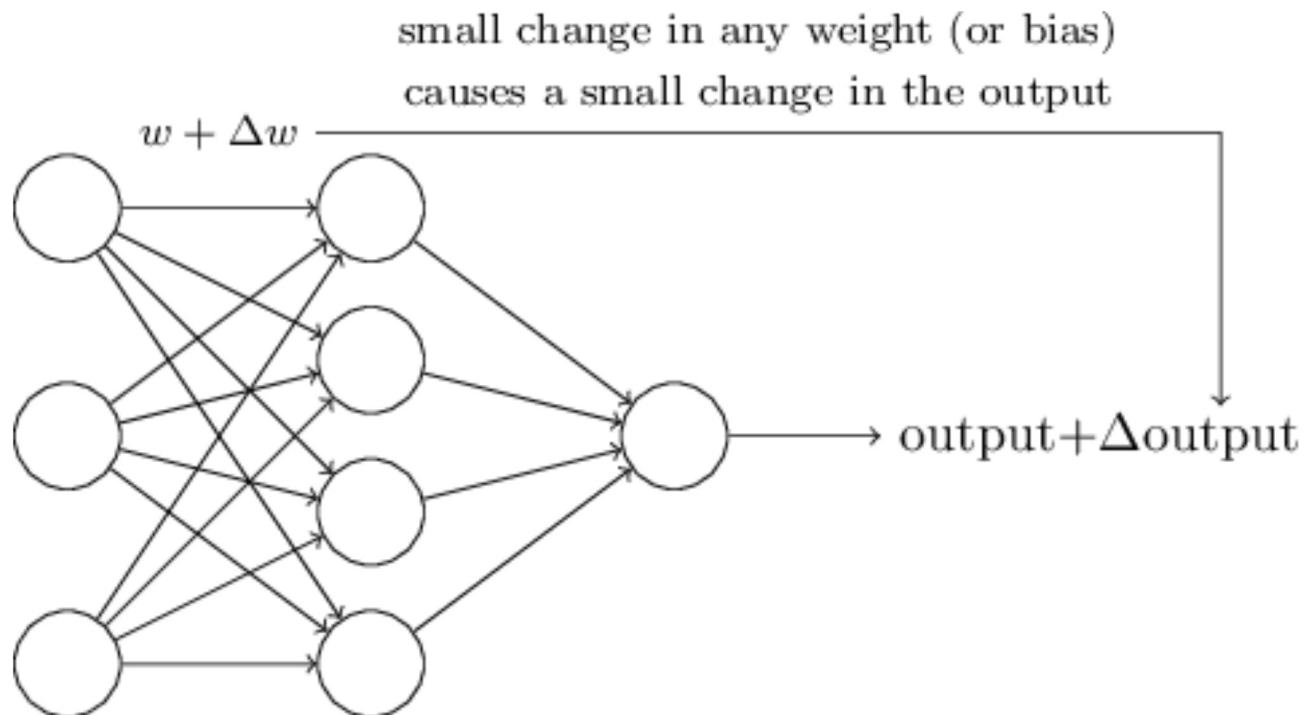
4. $\frac{d}{dx}(\ln x) = \frac{1}{x}$, for $x > 0$

5. $\frac{d}{dx}(\sin x) = \cos x$

6. $\frac{d}{dx}(\cos x) = -\sin x$

Error - loss

Vill kunna se hur diff påverkar fel



Loss vs. accuracy (noggrannhet)

- Noggrannhet i hur uppgiften utförs
 - T.ex. hur många katter som borde varit hundar
 - Allt över 0,5 räknas som 1 (dvs. korrekt)
 - Används för att utvärdera modeller (nätverk)
 - Lättare att jämföra modeller/uppgifter (t.ex. 94% korrekt klassificering)
- Loss
 - Mer precist, kontinuerligt mått
 - Används för att justera vikter
 - T.ex. MSE (Mean Squared Error)

För klassificering

- Uppgift: att plocka en klass (bland C möjliga klasser)
- Categorical Cross Entropy (Multinomial Logistic)
 - Mäter om sannolikhet för klass är korrekt (relativt hela ut-lagret)
 - Förutsätter softmax som akt.funktion i utlagret
- Sparse Categorical Cross Entropy
 - När väldigt många klasser, men bara en är korrekt
 - T.ex. ord i språk

För regression

- Uppgift: att producera ett exakt värde (t.ex. aktievärde)
- MAE (Mean Absolute Error)
- MSE (Mean Squared Error)
- MAPE (Mean Absolute Procentage Error)
 - Fel relativt target-storlek

rita.kovordanyi@liu.se

www.liu.se