

Deep Learning

Recurrent Neural Networks

RNN

Recurrent Neural Networks är designade för att hantera sekvenser av data, för ordningen spelar roll.

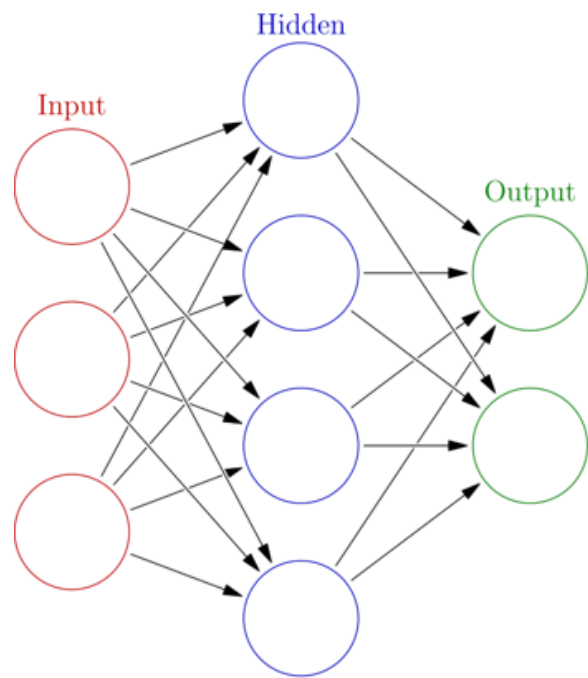
Meningen "Students cook delicious food" kan ses som en sekvens av ord.

"Delicious food cook students" har en annan mening.

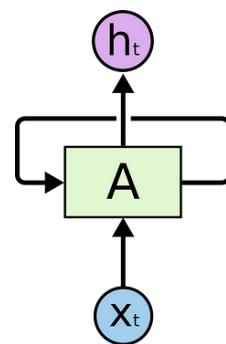
Vanliga nätverk kan ej hantera denna typ av data. Därför behöver vi en ny arkitektur.



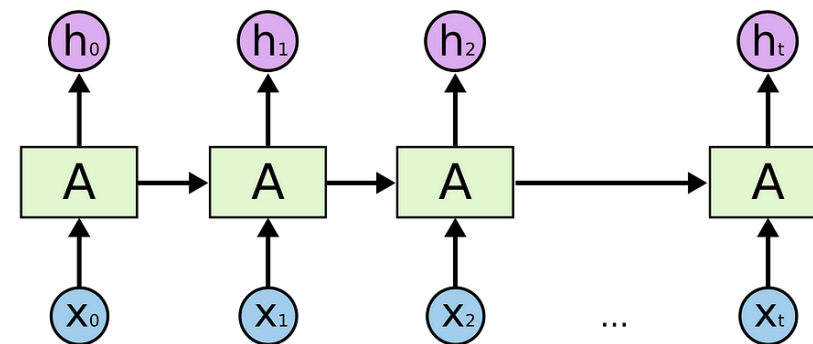
RNN



ANN



=



RNN

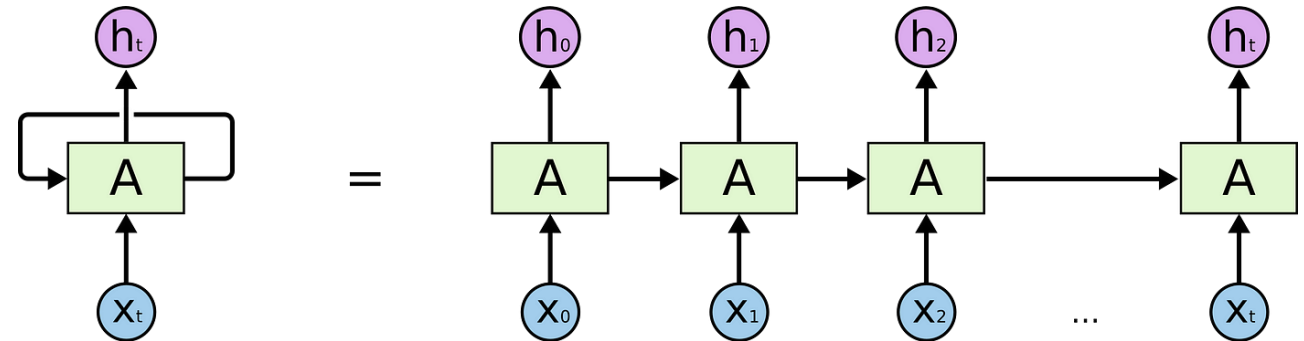
RNN

Beräkningen för ett tidssteg (exempelvis ett ord) har informationen om nuvarande tidssteg samt output från föregående tidssteg.

x_t är input

h_t är hidden state (minne)

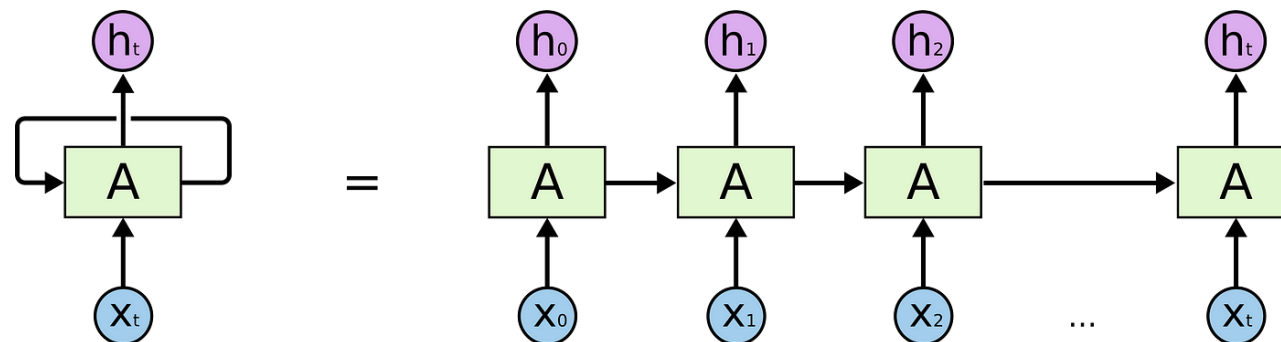
A är en updateringsfunktion



RNN

$$h_t = \sigma(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$

$$y_t = W_{hy}h_t + b_y$$



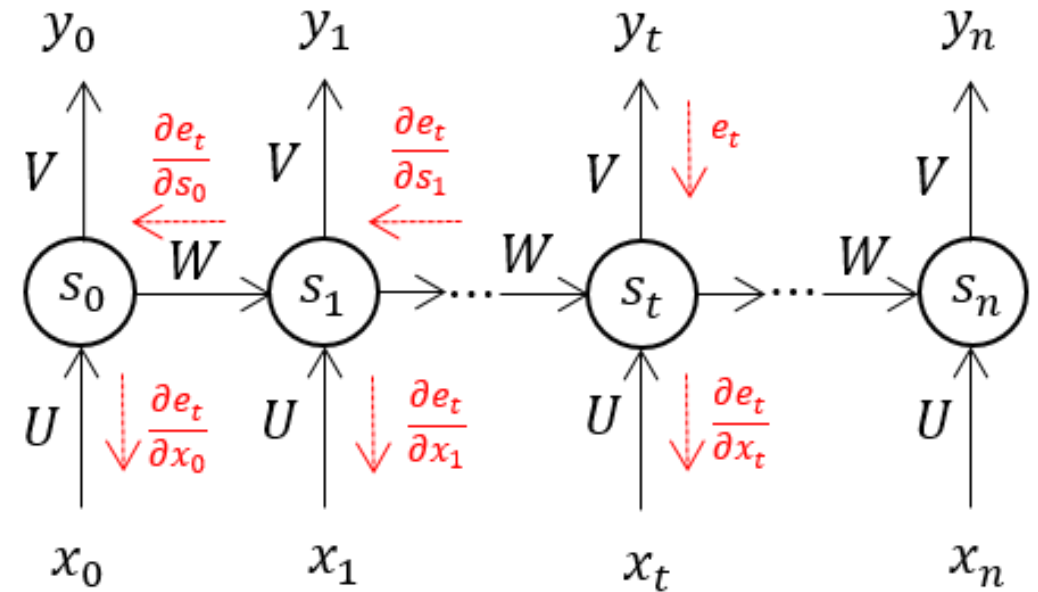
Bakåtpropagering (Backpropagation)

- Bakåtpropagering beräknar gradienten hos vikterna givet ett fel.
- “Felet” är skillnaden i vårt NNs output och vårt rätta svar, vår ground truth. (Skillnaden mäts genom loss funktionen)
- Med bakåtpropagering får vi reda på gradienten hos varje enskild vikt.
- Vi vill att alla vikters gradient ska hamna i globalt minima,
 - det betyder att felet på varje vikt är så litet det bara kan bli.
- Minimerar vi felet på alla vikter så har vi approximerat funktionen.

Backpropagation Through Time (BPTT)

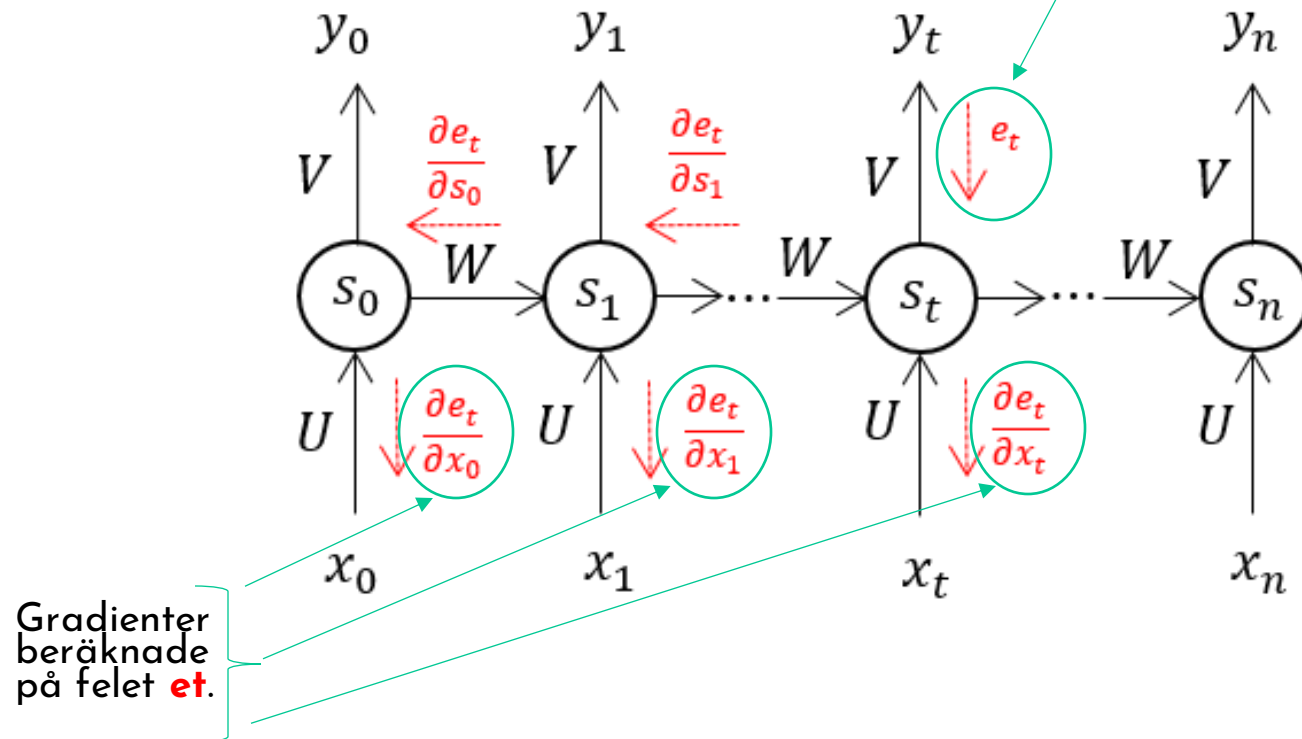
Precis som vanlig backpropagation används BPTT för att uppdatera vikter, men BPTT används i RNN.

Vid varje tidssteg beräknas en gradient. När alla tidssteg är gjorda så summeras gradienten för varje tidssteg och uppdaterar vikterna.



Backpropagation Through Time (BPTT)

Vid tidssteg t bräknas felet **e_t** . Det felet används sedan vid varje tidigare tidssteg för att beräkna gradienten.



RNN

Nu ska vi använda RNN för att göra en sentimentanalys.

Sentimentanalys innebär att vi försöker få ut en känsla från en text.

I vårt fall vill vi få ut om en text om en produkt är positiv, neutral eller negativ.



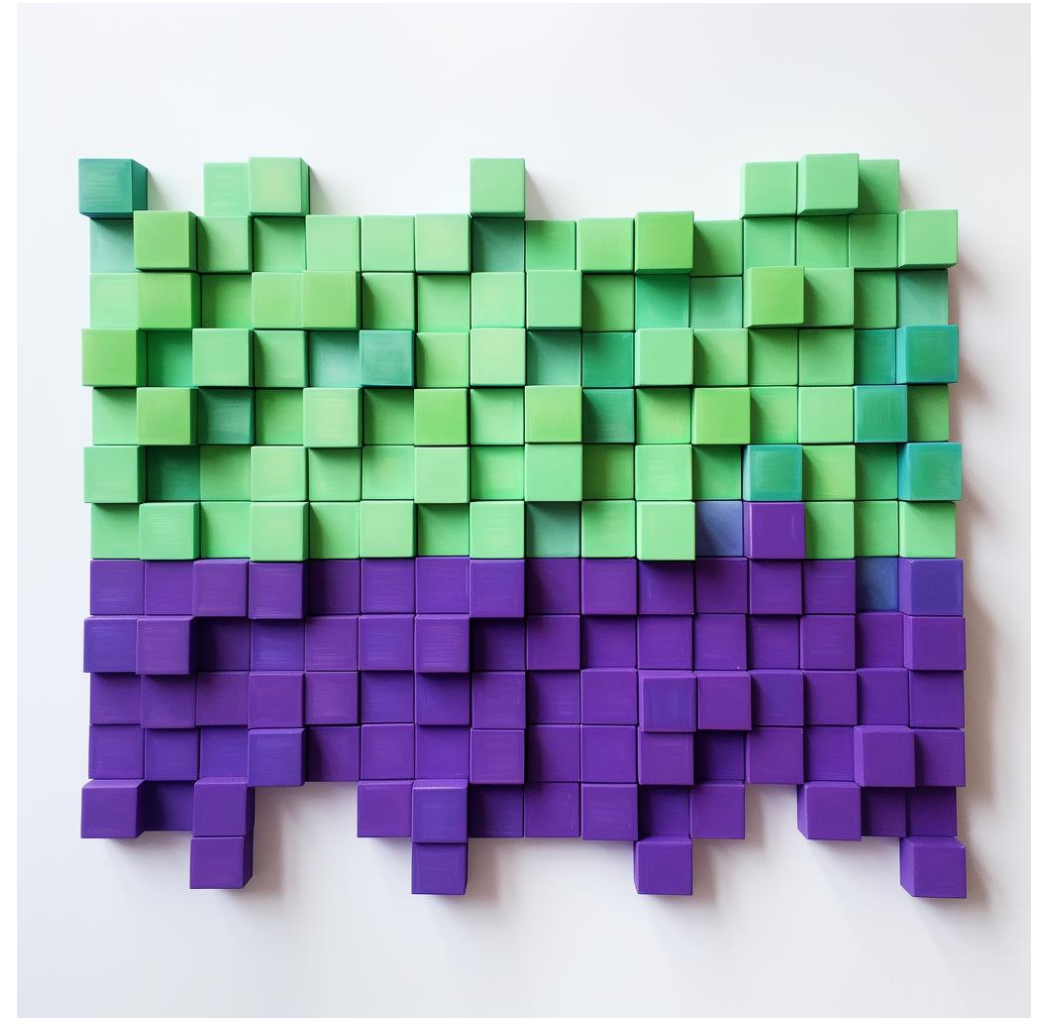
Deep Learning

Encoder-Decoder

Encoder-Decoder

Sequence-to-sequence (seq2seq) är en del inom machine learning där både input och output är sekvenser (till skillnad från sentimentanalysen vi gjorde tidigare).

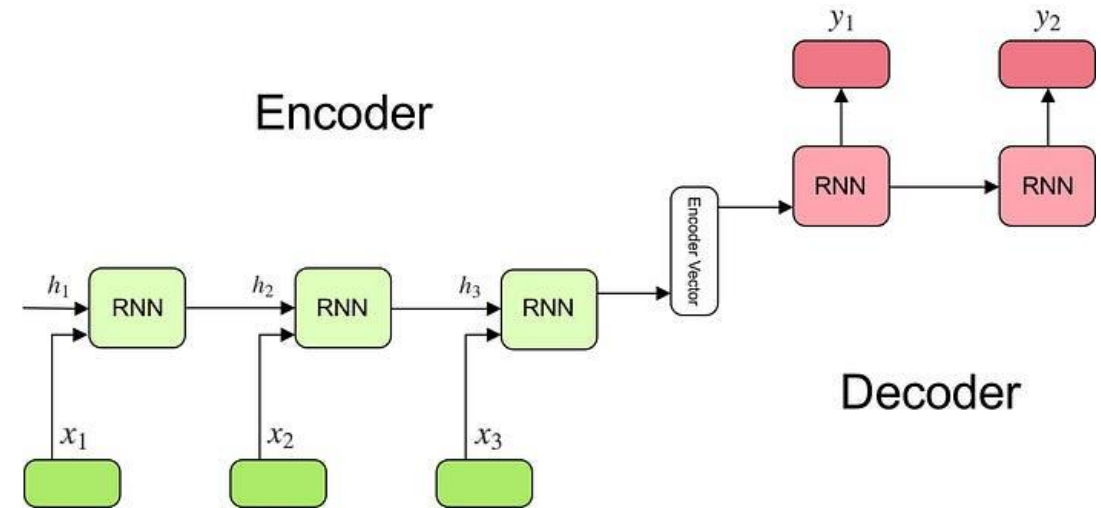
I denna kategori ingår chatbotar, översättning, summering av text, m.m.



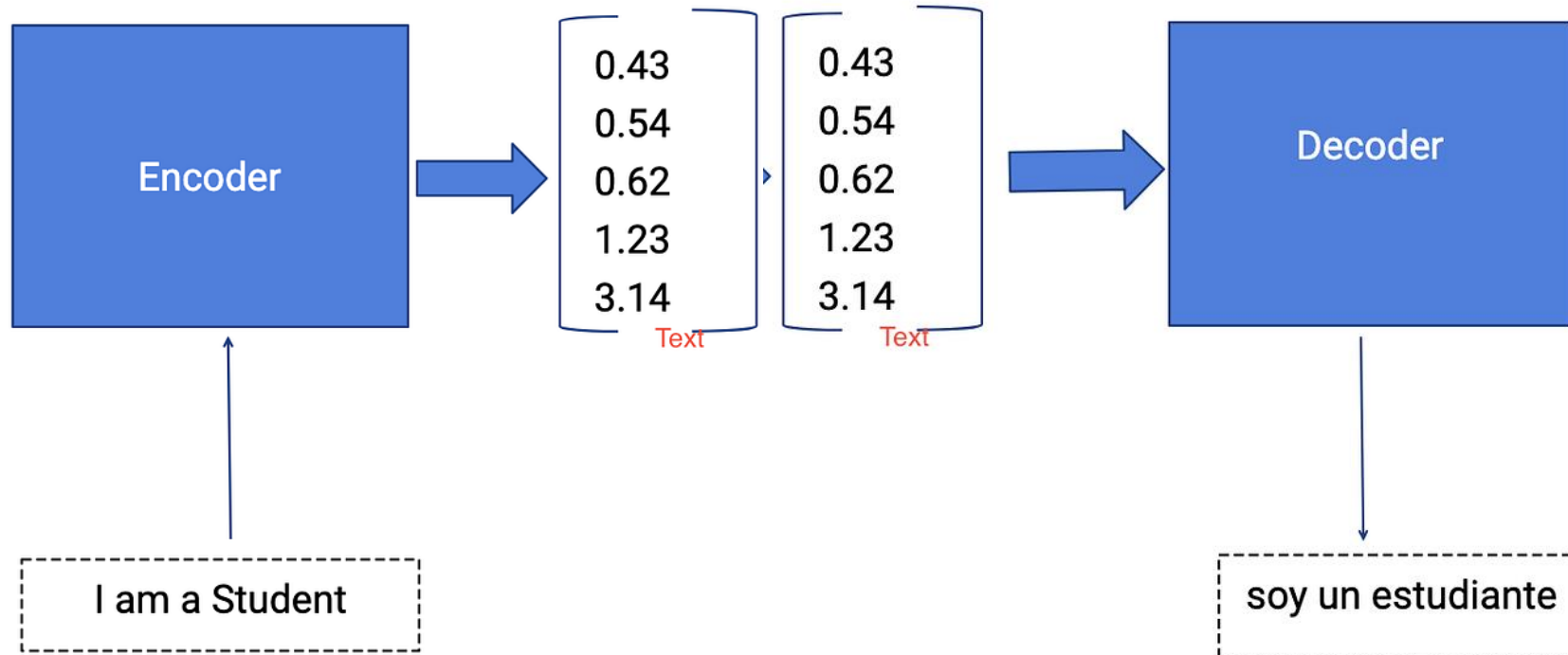
Encoder-Decoder

För att hantera sekvenser som input och sekvenser som output kan vi använda oss av en Encoder-Decoder arkitektur.

- Encodern tar en sekvens som input och ger ut en representation av fixed längd.
- Decodern tar en representation som input och ger ut en sekvens som output.



Encoder-Decoder



Deep Learning

Tidsserier

Tidsserier

Även tidsserier är sekvenser av data (precis som meningar).

Därför kan vi även använda RNN på den typen av data.

Tidsseriedata är data som har en följd och följden bestäms av vilken tidpunkt något skett.

Exempel på tidsseriedata:

- Väder
- Börsen
- Hälsodata



Tidsserier

När vi jobbar med tidsserier kan vi göra (och bör göra) dataprocessring som vi gör när vi jobbar med "vanlig data", men vi behöver hålla koll på ordningen.

- Feature engineering
- Hantera saknade värden
- Normalisering
- ...

