# Fundamentals of Machine Learning

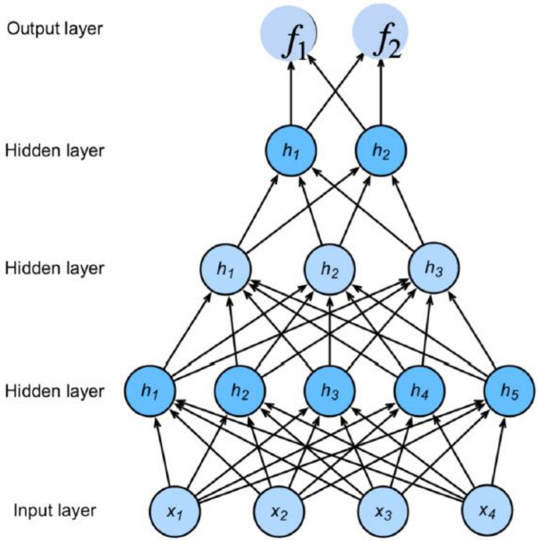
## NETWORK ARCHITECTURES, ATTENTION

Amit K Roy-Chowdhury

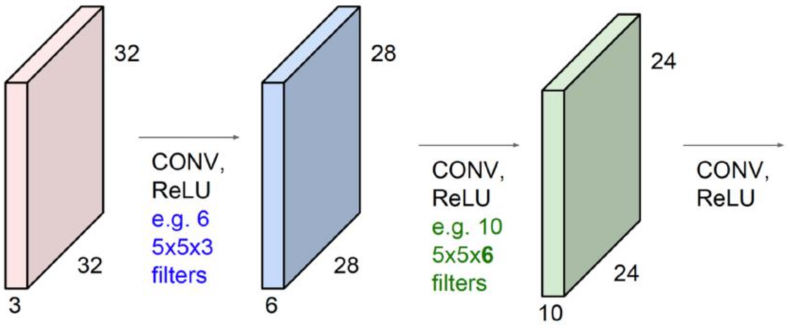
https://developer.ibm.com/articles/cc-machine-learning-deep-learning-architectures/

EE/CS 228 – Yinglun Zhu

# Feedforward Networks

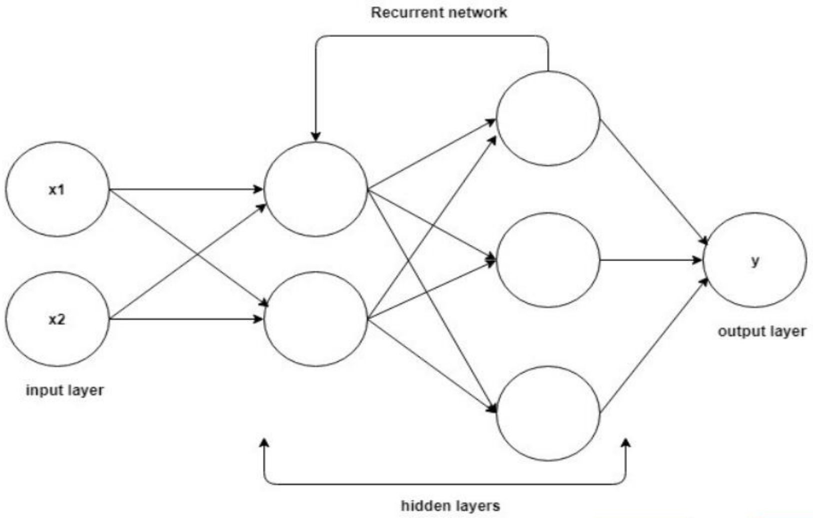


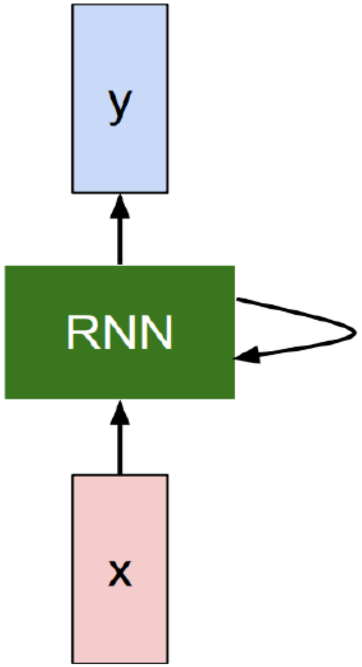
MLP



CNN

# Recurrent Neural Networks





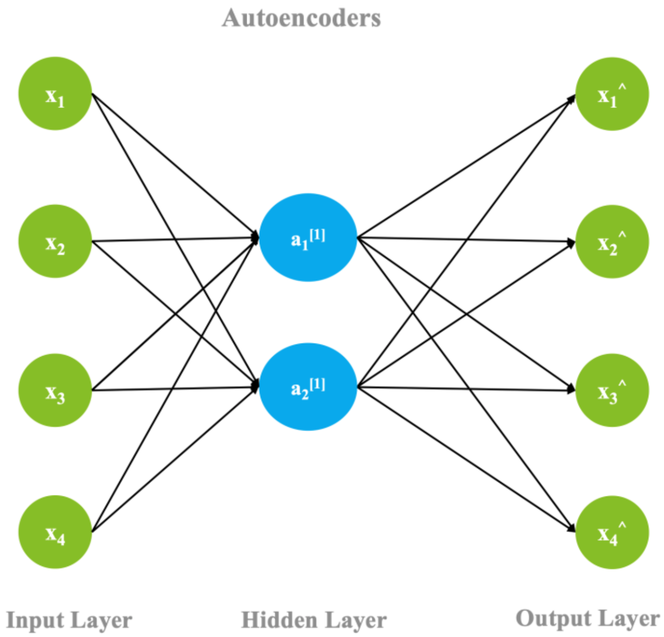
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

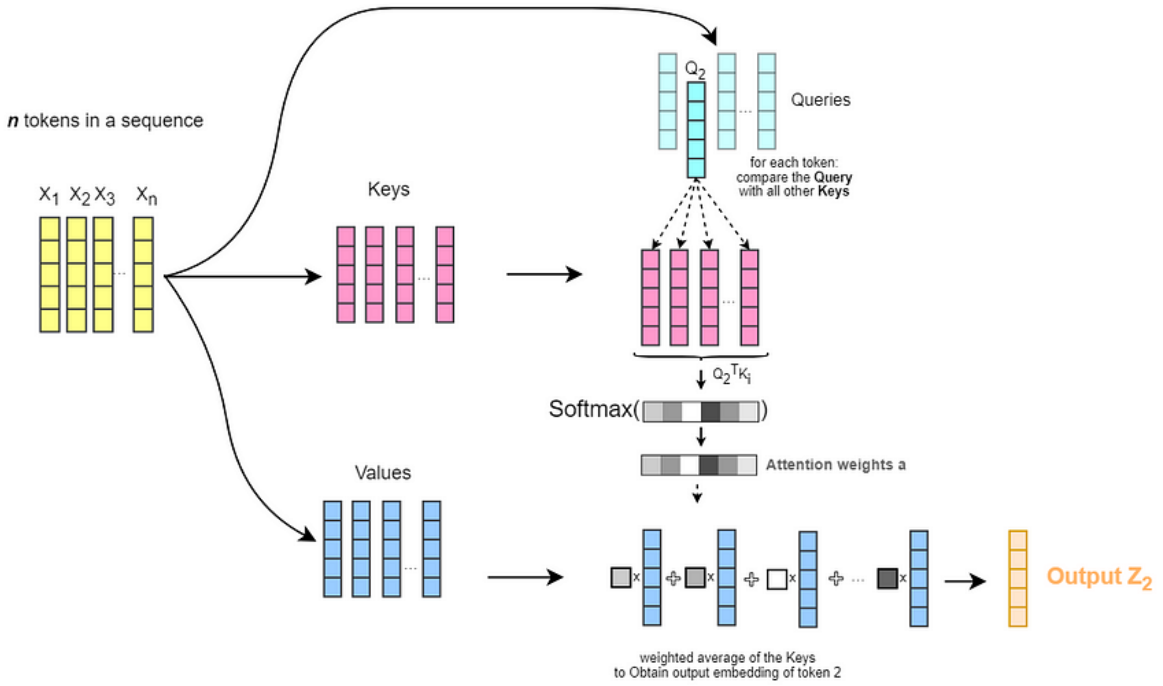
old state

input at some time step

# Autoencoders



# Transformers



https://towardsdatascience.com/what-are-query-key-and-value-in-the-transformer-architecture-and-why-are-they-used-acbe73f731f2

# Key, Query, Value

Value: $\mathbf{V} \in \mathbb{R}^{m \times v}$ set of $m$ feature vectors

Query: $\boldsymbol{q} \in \mathbb{R}^{q}$

Keys: $\mathbf{K} \in \mathbb{R}^{m \times k}$ set of $m$ **keys**

Find the query that is most similar to a key and use the corresponding value.

$$\mathrm{Attn}(\boldsymbol{q}, (\boldsymbol{k}_1, \boldsymbol{v}_1), \ldots, (\boldsymbol{k}_m, \boldsymbol{v}_m)) = \mathrm{Attn}(\boldsymbol{q}, (\boldsymbol{k}_{1:m}, \boldsymbol{v}_{1:m})) = \sum_{i=1}^{m} \alpha_i(\boldsymbol{q}, \boldsymbol{k}_{1:m}) \boldsymbol{v}_i$$

$$0 \leq \alpha_i(\boldsymbol{q}, \boldsymbol{k}_{1:m}) \leq 1 \qquad \sum_i \alpha_i(\boldsymbol{q}, \boldsymbol{k}_{1:m}) = 1$$

# Computing Attention Weights

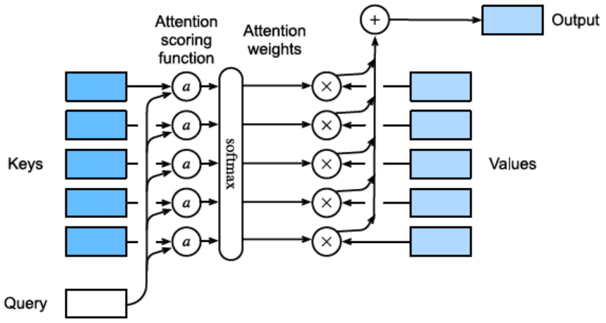
Find the query that is most similar to a key and use the corresponding value.

$$\mathrm{Attn}(\boldsymbol{q}, (\boldsymbol{k}_1, \boldsymbol{v}_1), \ldots, (\boldsymbol{k}_m, \boldsymbol{v}_m)) = \mathrm{Attn}(\boldsymbol{q}, (\boldsymbol{k}_{1:m}, \boldsymbol{v}_{1:m})) = \sum_{i=1}^{m} \alpha_i(\boldsymbol{q}, \boldsymbol{k}_{1:m}) \boldsymbol{v}_i$$

$$0 \leq \alpha_i(\boldsymbol{q}, \boldsymbol{k}_{1:m}) \leq 1 \qquad \textstyle\sum_i \alpha_i(\boldsymbol{q}, \boldsymbol{k}_{1:m}) = 1$$

Attention similarity/score: $a(\boldsymbol{q}, \boldsymbol{k}_i) \in \mathbb{R}$

$$\alpha_i(\boldsymbol{q}, \boldsymbol{k}_{1:m}) = \mathrm{softmax}_i([a(\boldsymbol{q}, \boldsymbol{k}_1), \ldots, a(\boldsymbol{q}, \boldsymbol{k}_m)]) = \frac{\exp(a(\boldsymbol{q}, \boldsymbol{k}_i))}{\sum_{j=1}^{m} \exp(a(\boldsymbol{q}, \boldsymbol{k}_j))}$$

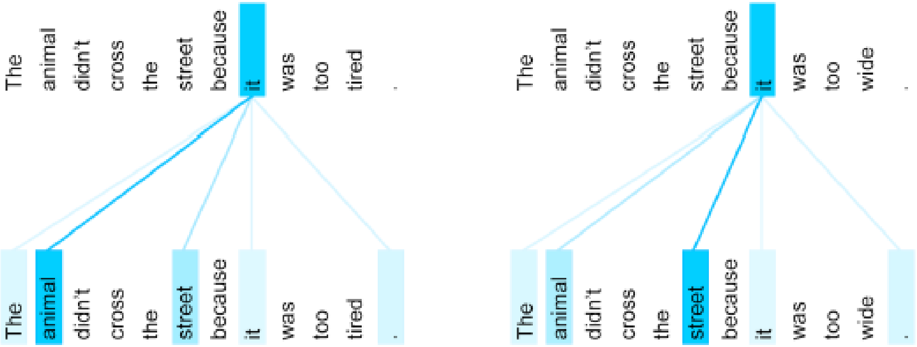


# Self Attention

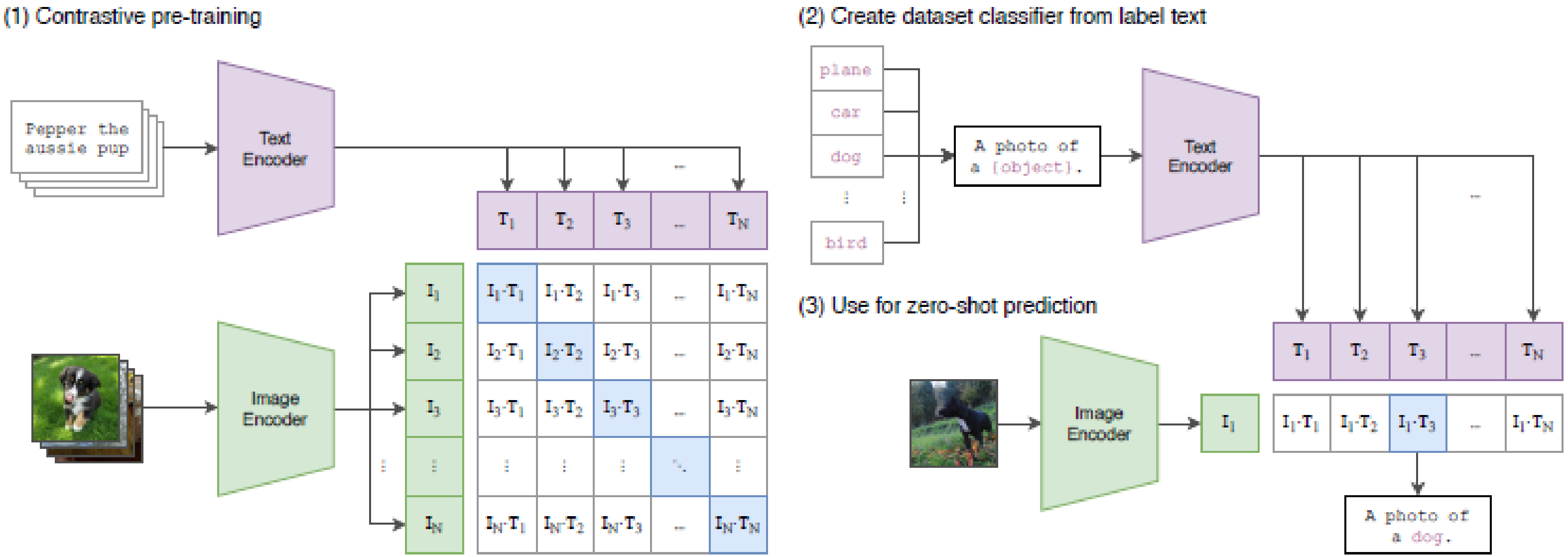
$$y_i = \mathrm{Attn}(x_i, (x_1, x_1), \ldots, (x_n, x_n))$$

input tokens $x_1, \ldots, x_n$, where $x_i \in \mathbb{R}^d$

query is $x_i$, and the keys and values are all the (valid) inputs $x_1, \ldots, x_n$



# Multimodal Models


(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
$T_1$ $T_2$ $T_3$ ... $T_N$
$I_1$ $I_2$ $I_3$ ⋮ $I_N$
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a [object].
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
$I_1$
$T_1$ $T_2$ $T_3$ ... $T_N$
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
A photo of a dog.