

Attention via $\log \sum \exp$ energy

Alexander Tschantz

January 21, 2025

1 General Framework

Setup. We consider a single set of nodes $\mathbf{v} = \{\mathbf{v}_a : a \in \{1, 2, \dots, A\}\}$, where each node $\mathbf{v}_a \in \mathbb{R}^d$. The relationships between these nodes are defined by a set of M energy functions $\{E_m : m \in \{1, 2, \dots, M\}\}$. Each energy function E_m defines a subset of nodes acting as *children* $C_m \subseteq \{1, 2, \dots, A\}$ and a subset acting as *parents* $P_m \subseteq \{1, 2, \dots, A\}$, which may overlap.

Energy. Each energy function E_m defines a similarity function:

$$\text{sim}(\mathbf{v}_c, \mathbf{v}_p) \quad : \quad \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}, \quad (1)$$

which produces a scalar similarity between a child \mathbf{v}_c and a parent \mathbf{v}_p . Using $\{\mathbf{v}_c\} = \{\mathbf{v}_c : c \in C_m\}$ and $\{\mathbf{v}_p\} = \{\mathbf{v}_p : p \in P_m\}$, the energy for E_m is defined as:

$$E_m(\{\mathbf{v}_c\}, \{\mathbf{v}_p\}) = - \sum_{c \in C_m} \ln \left(\sum_{p \in P_m} \exp(\text{sim}(\mathbf{v}_c, \mathbf{v}_p)) \right). \quad (2)$$

The global energy sums over all energy functions:

$$E(\{\mathbf{v}\}) = \sum_{m=1}^M E_m(\{\mathbf{v}_c\}, \{\mathbf{v}_p\}). \quad (3)$$

Gradient Updates. For a single node \mathbf{v}_a , the gradient of the global energy E w.r.t. \mathbf{v}_a decomposes into two terms. Let $\mathcal{M}_c(a) = \{m : a \in C_m\}$ denote the energy functions where \mathbf{v}_a acts as a *child*, and $\mathcal{M}_p(a) = \{m : a \in P_m\}$ the energy functions where \mathbf{v}_a acts as a *parent*. Then:

$$\begin{aligned} -\frac{\partial E}{\partial \mathbf{v}_a} = & \underbrace{\sum_{m \in \mathcal{M}_c(a)} \sum_{p \in P_m} \text{softmax}_p(\text{sim}(\mathbf{v}_a, \mathbf{v}_p)) \frac{\partial}{\partial \mathbf{v}_a} \text{sim}(\mathbf{v}_a, \mathbf{v}_p)}_{\mathbf{v}_a \text{ acting as a child}} \\ & + \underbrace{\sum_{m \in \mathcal{M}_p(a)} \sum_{c \in C_m} \text{softmax}_a(\text{sim}(\mathbf{v}_c, \mathbf{v}_a)) \frac{\partial}{\partial \mathbf{v}_a} \text{sim}(\mathbf{v}_c, \mathbf{v}_a)}_{\mathbf{v}_a \text{ acting as a parent}}. \end{aligned} \quad (4)$$

The first term captures contributions from \mathbf{v}_a being explained by its parents, while the second term captures contributions from \mathbf{v}_a explaining its children.

2 Gaussian Mixture Models

Setup. We have N data points (children) $\mathbf{x}_i \in \mathbb{R}^d$, $i \in C = \{1, \dots, N\}$, and K mixture components (parents), each with mean $\boldsymbol{\mu}_k \in \mathbb{R}^d$ and covariance $\boldsymbol{\Sigma}_k$, $k \in P = \{1, \dots, K\}$. Let π_k be the mixing proportion.

Similarity function. We define

$$\text{sim}(\mathbf{x}_i, \boldsymbol{\mu}_k) = \ln \pi_k - \frac{1}{2} (\mathbf{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k).$$

Energy.

$$E^{\text{GMM}}(\{\mathbf{x}_i\}, \{\boldsymbol{\mu}_k\}) = - \sum_{i=1}^N \ln \left(\sum_{k=1}^K \exp(\text{sim}(\mathbf{x}_i, \boldsymbol{\mu}_k)) \right). \quad (5)$$

Gradients. If we differentiate w.r.t. $\boldsymbol{\mu}_k$, then

$$-\frac{\partial E^{\text{GMM}}}{\partial \boldsymbol{\mu}_k} = \sum_{i=1}^N \text{softmax}_k(\mathbf{A}_{ik}) \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k).$$

Setting this gradient to zero yields the usual GMM M-step:

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^N \text{softmax}_k(\mathbf{A}_{ik}) \mathbf{x}_i}{\sum_{i=1}^N \text{softmax}_k(\mathbf{A}_{ik})}.$$

3 Cross Attention

Setup. We have a set of child vectors (queries) $\mathbf{Q} \in \mathbb{R}^{d \times N_Q}$ and a set of parent vectors (keys) $\mathbf{K} \in \mathbb{R}^{d \times N_K}$. Let

$$C = \{1, \dots, N_Q\}, \quad P = \{1, \dots, N_K\},$$

so $\mathbf{v}_c = \mathbf{q}_c$ is the c -th query, and $\mathbf{v}_p = \mathbf{k}_p$ is the p -th key. Suppose we have learnable weight matrices $\mathbf{W}^Q, \mathbf{W}^K \in \mathbb{R}^{d \times d}$. Then

$$\mathbf{q}_c = \mathbf{W}^Q \mathbf{x}_c^Q, \quad \mathbf{k}_p = \mathbf{W}^K \mathbf{x}_p^K,$$

where \mathbf{x}_c^Q is the raw c -th query token and \mathbf{x}_p^K the raw p -th key token.

Similarity function.

$$\text{sim}(\mathbf{q}_c, \mathbf{k}_p) = \mathbf{q}_c^\top \mathbf{k}_p.$$

Energy.

$$E^{\text{Cross}}(\{\mathbf{q}_c\}, \{\mathbf{k}_p\}) = - \sum_{c=1}^{N_Q} \ln \left(\sum_{p=1}^{N_K} \exp(\mathbf{q}_c^\top \mathbf{k}_p) \right). \quad (6)$$

Gradients.

$$-\frac{\partial E^{\text{Cross}}}{\partial \mathbf{q}_c} = \sum_{p=1}^{N_K} \text{softmax}_p(\mathbf{q}_c^\top \mathbf{k}_p) \mathbf{k}_p. \quad (7)$$

$$-\frac{\partial E^{\text{Cross}}}{\partial \mathbf{k}_p} = \sum_{c=1}^{N_Q} \text{softmax}_p(\mathbf{q}_c^\top \mathbf{k}_p) \mathbf{q}_c. \quad (8)$$

When mapping back to the raw tokens \mathbf{x}_c^Q or \mathbf{x}_p^K , chain-rule multiplies by \mathbf{W}^Q or \mathbf{W}^K , respectively.

4 Hopfield Networks

Setup. We have a set of *children* data vectors $\mathbf{x}_i \in \mathbb{R}^d, i \in C = \{1, \dots, N\}$, and a set of *parent* memory vectors $\mathbf{m}_\mu \in \mathbb{R}^d, \mu \in P = \{1, \dots, K\}$.

Similarity function.

$$\text{sim}(\mathbf{x}_i, \mathbf{m}_\mu) = \mathbf{x}_i^\top \mathbf{m}_\mu.$$

Energy.

$$E^{\text{Hopfield}}(\{\mathbf{x}_i\}, \{\mathbf{m}_\mu\}) = - \sum_{i=1}^N \ln \left(\sum_{\mu=1}^K \exp(\mathbf{x}_i^\top \mathbf{m}_\mu) \right). \quad (9)$$

Gradients.

$$-\frac{\partial E^{\text{Hopfield}}}{\partial \mathbf{x}_i} = \sum_{\mu=1}^K \text{softmax}_{\mu}(\mathbf{x}_i^{\top} \mathbf{m}_{\mu}) \mathbf{m}_{\mu}. \quad (10)$$

$$-\frac{\partial E^{\text{Hopfield}}}{\partial \mathbf{m}_{\mu}} = \sum_{i=1}^N \text{softmax}_{\mu}(\mathbf{x}_i^{\top} \mathbf{m}_{\mu}) \mathbf{x}_i. \quad (11)$$

5 Slot Attention

Setup. Let $\mathbf{x}_j \in \mathbb{R}^d$, $j \in C = \{1, \dots, N\}$ be the children (tokens), and $\boldsymbol{\mu}_i \in \mathbb{R}^d$, $i \in P = \{1, \dots, S\}$ be the parents (slots). We typically apply linear transforms $\mathbf{W}_K, \mathbf{W}_Q \in \mathbb{R}^{d \times d}$ to form

$$\text{sim}(\mathbf{x}_j, \boldsymbol{\mu}_i) = (\mathbf{W}_K \mathbf{x}_j)^{\top} (\mathbf{W}_Q \boldsymbol{\mu}_i).$$

Energy.

$$E^{\text{Slot}}(\{\mathbf{x}_j\}, \{\boldsymbol{\mu}_i\}) = -\sum_{j=1}^N \ln \left(\sum_{i=1}^S \exp(\text{sim}(\mathbf{x}_j, \boldsymbol{\mu}_i)) \right). \quad (12)$$

Gradients.

$$-\frac{\partial E^{\text{Slot}}}{\partial \mathbf{x}_j} = \sum_{i=1}^S \text{softmax}_i(\text{sim}(\mathbf{x}_j, \boldsymbol{\mu}_i)) \mathbf{W}_K^{\top} \mathbf{W}_Q \boldsymbol{\mu}_i. \quad (13)$$

$$-\frac{\partial E^{\text{Slot}}}{\partial \boldsymbol{\mu}_i} = \sum_{j=1}^N \text{softmax}_i(\text{sim}(\mathbf{x}_j, \boldsymbol{\mu}_i)) \mathbf{W}_Q^{\top} \mathbf{W}_K \mathbf{x}_j. \quad (14)$$

6 Self-Attention

Setup. In self-attention, every node can act as both a child (query) and a parent (key). Concretely, let us have N tokens $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$. We form

$$\mathbf{q}_i = \mathbf{W}^Q \mathbf{x}_i, \quad \mathbf{k}_i = \mathbf{W}^K \mathbf{x}_i,$$

for $i = 1, \dots, N$. Thus the set $C = \{1, \dots, N\}$ and $P = \{1, \dots, N\}$ coincide, with

$$\text{sim}(\mathbf{x}_c, \mathbf{x}_p) = (\mathbf{W}^Q \mathbf{x}_c)^{\top} (\mathbf{W}^K \mathbf{x}_p).$$

Energy.

$$E^{\text{SA}}(\{\mathbf{x}_i\}) = -\sum_{c=1}^N \ln \left(\sum_{p=1}^N \exp \left((\mathbf{W}^Q \mathbf{x}_c)^{\top} (\mathbf{W}^K \mathbf{x}_p) \right) \right). \quad (15)$$

Gradients. Since each \mathbf{x}_i is *both* a child and a parent, its gradient is a sum of two terms (the child side and the parent side). Writing it out explicitly:

$$\begin{aligned} -\frac{\partial E^{\text{SA}}}{\partial \mathbf{x}_i} &= \underbrace{\sum_{p=1}^N \text{softmax}_p \left((\mathbf{W}^Q \mathbf{x}_i)^{\top} (\mathbf{W}^K \mathbf{x}_p) \right) \mathbf{W}_Q^{\top} \mathbf{W}_K \mathbf{x}_p}_{\text{child } i \text{ being explained by parents } p} \\ &\quad + \underbrace{\sum_{c=1}^N \text{softmax}_i \left((\mathbf{W}^Q \mathbf{x}_c)^{\top} (\mathbf{W}^K \mathbf{x}_i) \right) \mathbf{W}_K^{\top} \mathbf{W}_Q \mathbf{x}_c}_{\text{parent } i \text{ explaining children } c}. \end{aligned} \quad (16)$$

7 Coordinate ascent

Consider an energy of the form

$$E(\{\mathbf{x}_j\}, \{\boldsymbol{\mu}_i\}; \theta) = - \sum_{j=1}^N \ln \left(\sum_{i=1}^S \exp(\text{sim}_\theta(\mathbf{x}_j, \boldsymbol{\mu}_i)) \right),$$

with θ denoting the parameters of the similarity function. In an EM procedure, we alternate:

- **E-step:** Update latent variables using fixed θ .
- **M-step:** Update parameters θ in closed form given fixed latent assignments.

For *slot attention*, we have:

$$\begin{aligned} \theta &= \{\mathbf{W}_K, \mathbf{W}_Q\}, \\ \text{sim}_\theta(\mathbf{x}_j, \boldsymbol{\mu}_i) &= (\mathbf{W}_K \mathbf{x}_j)^\top (\mathbf{W}_Q \boldsymbol{\mu}_i). \\ \mathbf{A} \text{ with entries } A_{ji} &= \text{softmax}_i \left((\mathbf{W}_K \mathbf{x}_j)^\top (\mathbf{W}_Q \boldsymbol{\mu}_i) \right). \end{aligned}$$

E-step: Update Slots

Fix $\mathbf{W}_K, \mathbf{W}_Q$. The gradient for each slot $\boldsymbol{\mu}_i$ is

$$-\frac{\partial E^{\text{Slot}}}{\partial \boldsymbol{\mu}_i} = \sum_{j=1}^N A_{ji} \mathbf{W}_Q^\top \mathbf{W}_K \mathbf{x}_j.$$

Use this gradient to iteratively update $\{\boldsymbol{\mu}_i\}$ until convergence.

M-step: Update Parameters

Given fixed slots $\{\boldsymbol{\mu}_i\}$ and attention matrix \mathbf{A} , we aim to update the parameters $\mathbf{W}_K, \mathbf{W}_Q$. This is motivated by setting the gradient of the energy with respect to these parameters to zero:

$$-\frac{\partial E^{\text{Slot}}}{\partial \mathbf{W}_K} = 0, \quad -\frac{\partial E^{\text{Slot}}}{\partial \mathbf{W}_Q} = 0.$$

Under the fixed assignments provided by \mathbf{A} and slots $\{\boldsymbol{\mu}_i\}$, these conditions are equivalent to solving a weighted least-squares problem. Specifically, we consider minimizing the objective

$$\min_{\mathbf{W}_K, \mathbf{W}_Q} \sum_{j=1}^N \sum_{i=1}^S A_{ji} \left\| \mathbf{W}_K \mathbf{x}_j - \mathbf{W}_Q \boldsymbol{\mu}_i \right\|^2,$$

since the stationary point of this quadratic form corresponds to zero gradients with respect to $\mathbf{W}_K, \mathbf{W}_Q$.

Update \mathbf{W}_Q : Differentiate w.r.t. \mathbf{W}_Q , set to zero:

$$\sum_{j,i} A_{ji} \left(\mathbf{W}_K \mathbf{x}_j - \mathbf{W}_Q \boldsymbol{\mu}_i \right) \boldsymbol{\mu}_i^\top = 0,$$

yielding

$$\mathbf{W}_Q = \left(\sum_{j,i} A_{ji} \mathbf{W}_K \mathbf{x}_j \boldsymbol{\mu}_i^\top \right) \left(\sum_{j,i} A_{ji} \boldsymbol{\mu}_i \boldsymbol{\mu}_i^\top \right)^{-1}.$$

Update \mathbf{W}_K : Similarly, differentiate w.r.t. \mathbf{W}_K , set to zero:

$$\sum_{j,i} A_{ji} \left(\mathbf{W}_K \mathbf{x}_j - \mathbf{W}_Q \boldsymbol{\mu}_i \right) \mathbf{x}_j^\top = 0,$$

yielding

$$\mathbf{W}_K = \left(\sum_{j,i} A_{ji} \mathbf{W}_Q \boldsymbol{\mu}_i \mathbf{x}_j^\top \right) \left(\sum_{j,i} A_{ji} \mathbf{x}_j \mathbf{x}_j^\top \right)^{-1}.$$

Iterate: Alternate between the E-step (updating $\{\boldsymbol{\mu}_i\}$) and the M-step (updating $\mathbf{W}_K, \mathbf{W}_Q$) until convergence.

A Appendix: Derivation of Gradient Updates

Let us consider a generic term:

$$-\ln\left(\sum_{m=1}^M \exp(f_m(\mathbf{x}))\right),$$

where $\mathbf{x} \in \mathbb{R}^d$ is some variable, and each f_m is a scalar function. We compute its gradient:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}} \left[-\ln\left(\sum_{m=1}^M \exp(f_m(\mathbf{x}))\right) \right] &= -\frac{1}{\sum_{m'} \exp(f_{m'}(\mathbf{x}))} \sum_{m=1}^M \exp(f_m(\mathbf{x})) \frac{\partial f_m(\mathbf{x})}{\partial \mathbf{x}} \\ &= -\sum_{m=1}^M \left[\frac{\exp(f_m(\mathbf{x}))}{\sum_{m'} \exp(f_{m'}(\mathbf{x}))} \right] \frac{\partial f_m(\mathbf{x})}{\partial \mathbf{x}}. \end{aligned}$$

Defining $\text{softmax}_m(f(\mathbf{x})) = \frac{\exp(f_m(\mathbf{x}))}{\sum_{m'} \exp(f_{m'}(\mathbf{x}))}$, this is

$$-\sum_{m=1}^M \text{softmax}_m(f(\mathbf{x})) \frac{\partial f_m(\mathbf{x})}{\partial \mathbf{x}},$$

which matches the softmax-weighted gradient structure.