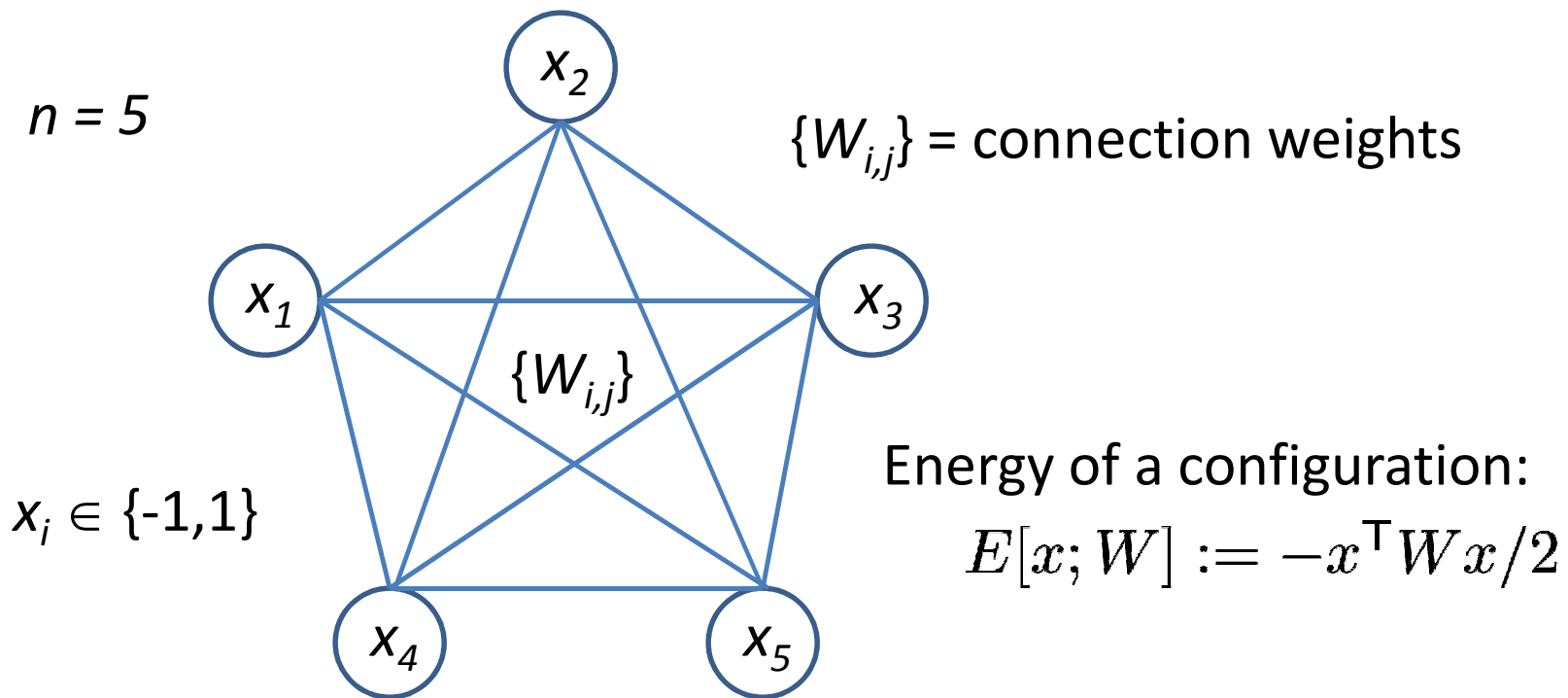


Restricted Boltzmann Machines: Learning, and Hardness of Inference

Presented by:
Nakul Verma

Recall: Hopfield Nets

- An energy based network of interconnected inputs (typically referred to as *neurons*) to encode memories.



Goal: to *encode* memories -- that is, given p configurations z^1, \dots, z^p , *learn* W such that each configuration z^i is (locally) a low energy state.

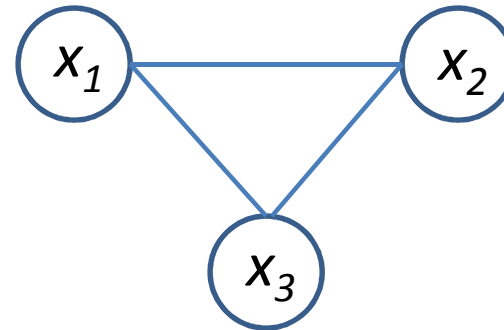
Hopfield Nets: limitations

- Not all memories can be encoded in this way!
Typically: Hopfield Nets cannot encode high-order correlations

Example

- Suppose $n = 3$ and we want to remember only two memories:
learn patterns: $z^1 = \{1, -1, 1\}$ and $z^2 = \{1, 1, -1\}$

Want to learn: parameters W in a Hopfield Net, that assign locally low energy to these memories



What W should we choose?

Hopfield Nets: limitations

- Recall, we assigned the weights as

$$W := \sum_i z^i (z^i)^\top$$

Memories to encode:

$$z^1 = \{1, -1, 1\}$$

$$z^2 = \{1, 1, -1\}$$

It turns out that this assignment does **NOT** assign low energy (locally) to the given memory configurations.

Why? In this case,
$$W = z^1(z^1)^\top + z^2(z^2)^\top = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & -2 \\ 0 & -2 & 2 \end{bmatrix}$$

so,
$$E[z^1; W] = E[z^2; W] = -5$$

BUT: for $x = \{-1, 1, -1\}$ (hamming distance 1 from z^2)

$$E[x; W] = -5$$

So, W is **not** locally energy minimizing!

Hopfield Nets: limitations

Question: Is there *any* symmetric W for which the memories z^1 and z^2 are (locally) energy minimizing configurations?

Unfortunately, the answer is still **NO**.

Solution:

Make use of hidden units (aka **Restricted Boltzmann Machines**)

Restricted Boltzmann Machines: An overview

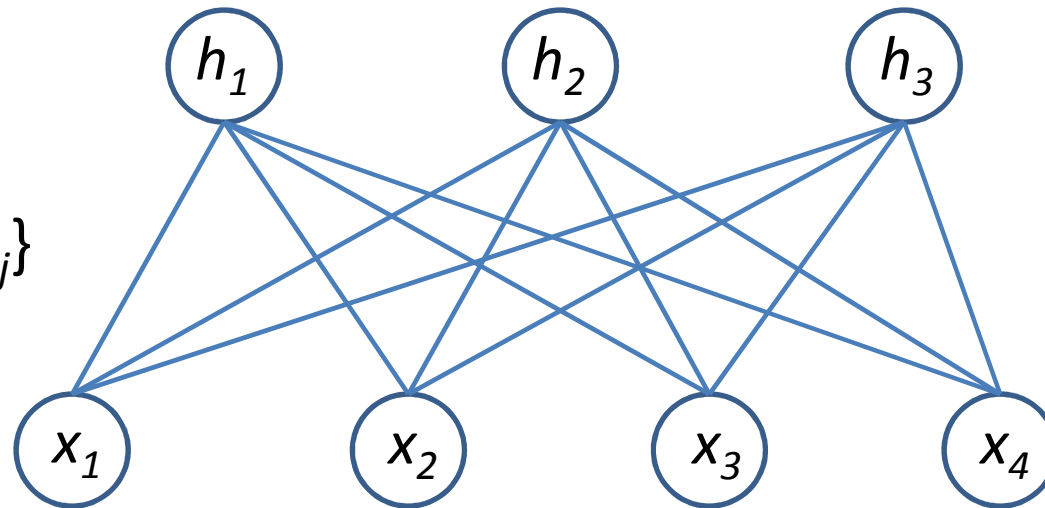
- A **bipartite** network between input and hidden variables
- Was introduced as:
 - ‘Harmoniums’ by Smolensky [Smo87]
 - ‘Influence Combination Machines’ by Freund and Haussler [FH91]
- Expressive enough to encode any distribution while being computationally efficient!

RBM: the structure

Hidden units:

weights $\{w_{i,j}\}$

Input units:



$m = 3$
 $h_i \in \{0, 1\}$

$n = 4$
 $x_j \in \{-1, +1\}$, or
 $x_j \in (-\infty, \infty)$

Energy of a configuration:

$$E[x, h; W] := -x^T W h \quad (\text{if } x \text{ binary})$$

$$E[x, h; W] := -x^T W h + \|x\|^2 \quad (\text{if } x \text{ real})$$

Probability of a state:

$$P[x|W] \propto \sum_{h \in \{0,1\}^m} e^{-E(x,h;W)}$$

$$= \prod_{i=1}^m (1 + e^{-x \cdot W_{:i}}) \quad (\text{if } x \text{ binary})$$

$$= e^{-\frac{1}{2} \|x\|^2} \prod_{i=1}^m (1 + e^{-x \cdot W_{:i}}) \quad (\text{if } x \text{ real})$$

RBM: What can we do?

- **Can** encode *any* distribution over $\{-1,1\}^n$!
well... given enough hidden units.
- **Can** estimate the right number of hidden units.
will use a variant of projection pursuit method.
- **Cannot** efficiently estimate the $P[x|W]$
cannot even approximate it!

Talk Outline

We will discuss each of the issues in detail.

- Universality [FH91].
- Learning the structure of RBM [FH91].
- Hardness of approximate inference [LS10].

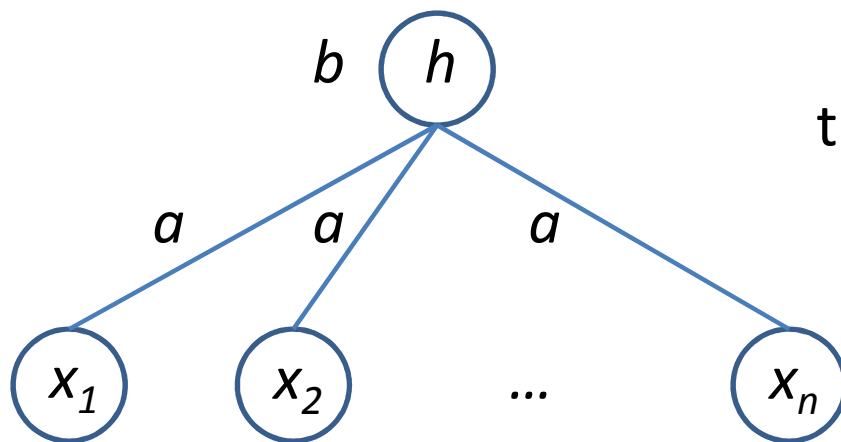
Talk Outline

- **Universality [FH91].**
- Learning the structure of RBM [FH91].
- Hardness of approximate inference [LS10].

RBM: Universality [FH91]

- Pick a configuration, say, $x = \{1, 1, \dots, 1\}$. Suppose we want to learn weights W in an RBM that assigns $P[x|W] = p$.
- How can we do that?

Consider



$$\text{if } a = \frac{1}{2} \ln(q - 1) + \frac{1}{2} \ln(1/\epsilon) \\ b = -na + \ln(q - 1)$$

then

$$f(x; a, b) = q \quad \text{if } x = \{1, \dots, 1\} \\ 1 \leq f(x; a, b) \leq 1 + \epsilon \quad \text{o.w.}$$

$$E[x, h; a, b] := -(a \sum_i x_i + b)h$$

$$P[x|a, b] \propto f(x; a, b) := 1 + e^{b+a \sum_i x_i}$$

RBM: Universality [FH91]

- Since we want $p = P[x|a, b] \approx \frac{q}{q+(2^n-1)}$ (for $x = \{1, \dots, 1\}$)

we want to set q to $\frac{p(2^n-1)}{1-p}$

- This can be easily generalized to **different probability assignments** for **different configurations** by adding additional hidden units.
- Therefore, in general, we can approximate any distribution by adding **sufficiently many** hidden units (2^n in the worst case)

Talk Outline

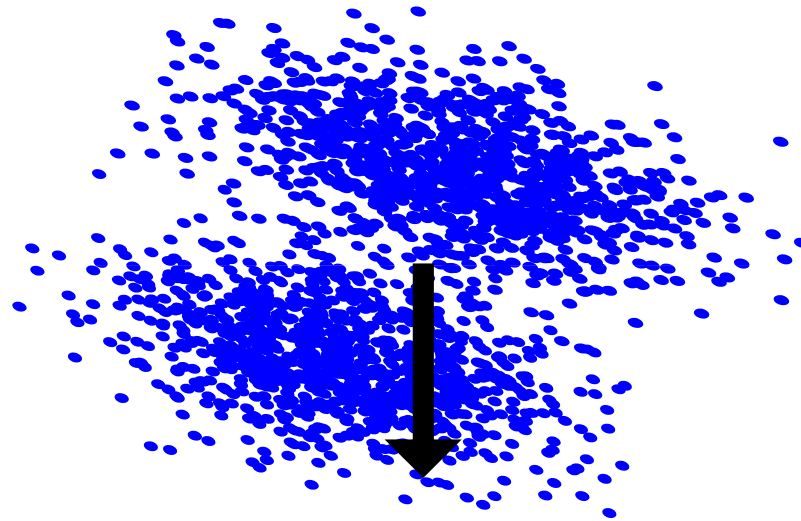
- Universality [FH91].
- Learning the structure of RBM [FH91].
- Hardness of approximate inference [LS10].

RBM: Learning the structure

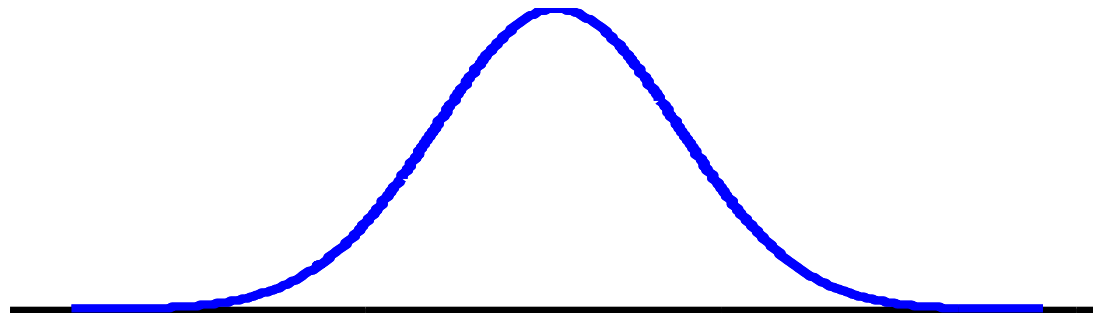
- Recall that each hidden unit is able to encode at least one high-order interaction among the input variables.
- During learning, we want to have as **few hidden units** as possible while maintaining good representation.
 - Having too few units will give poor prediction, while having too many units will overfit the training data.
- Question: how can we estimate the **right number** of units?
Solution: projection pursuit!

Detour: Projection Pursuit (PP)

- A methodology for finding **interesting characteristics** of your underlying data distribution by observing **1D projections** [FT74]

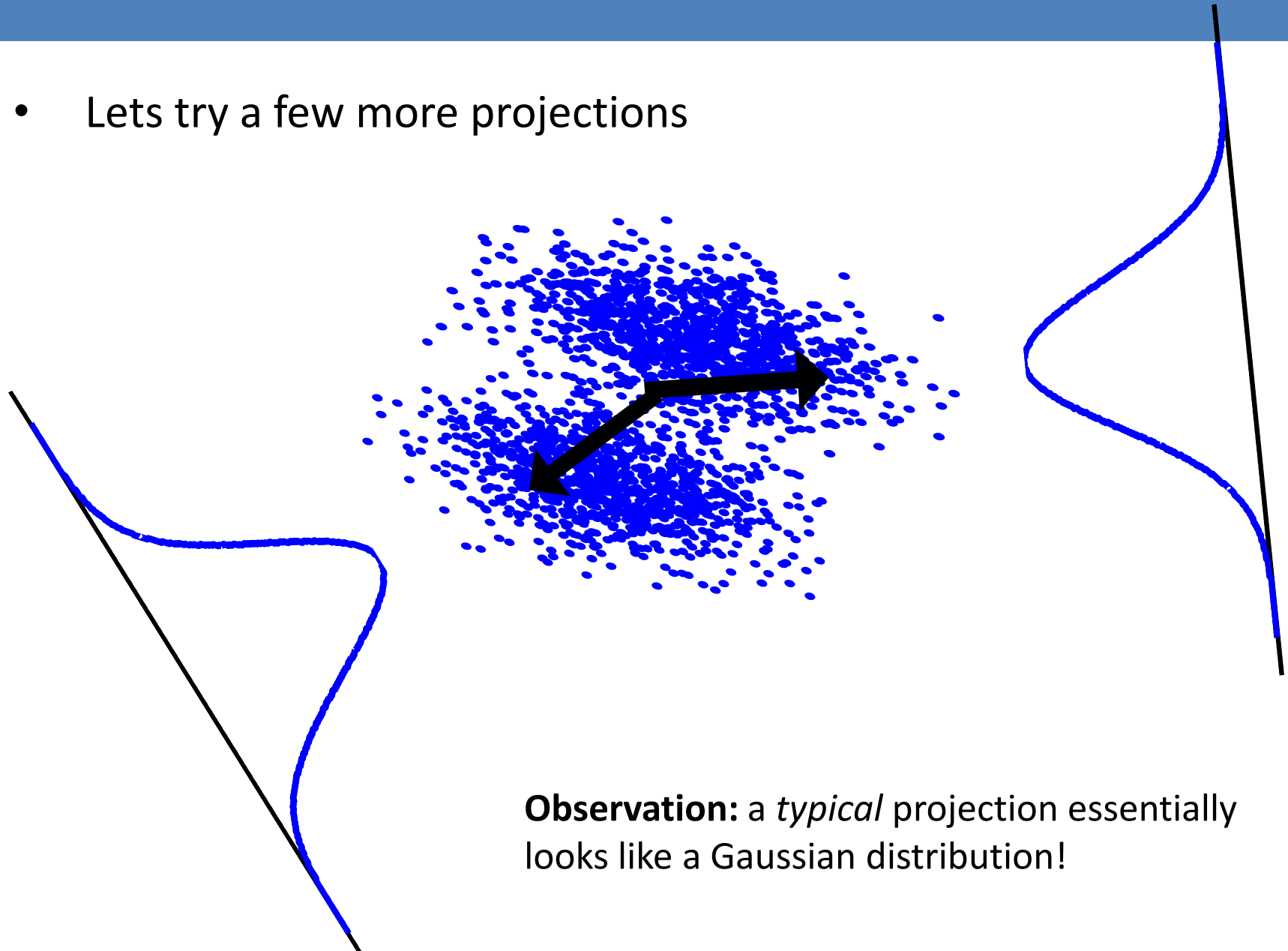


pick a projection
direction, say, ↓



Detour: Projection Pursuit (PP)

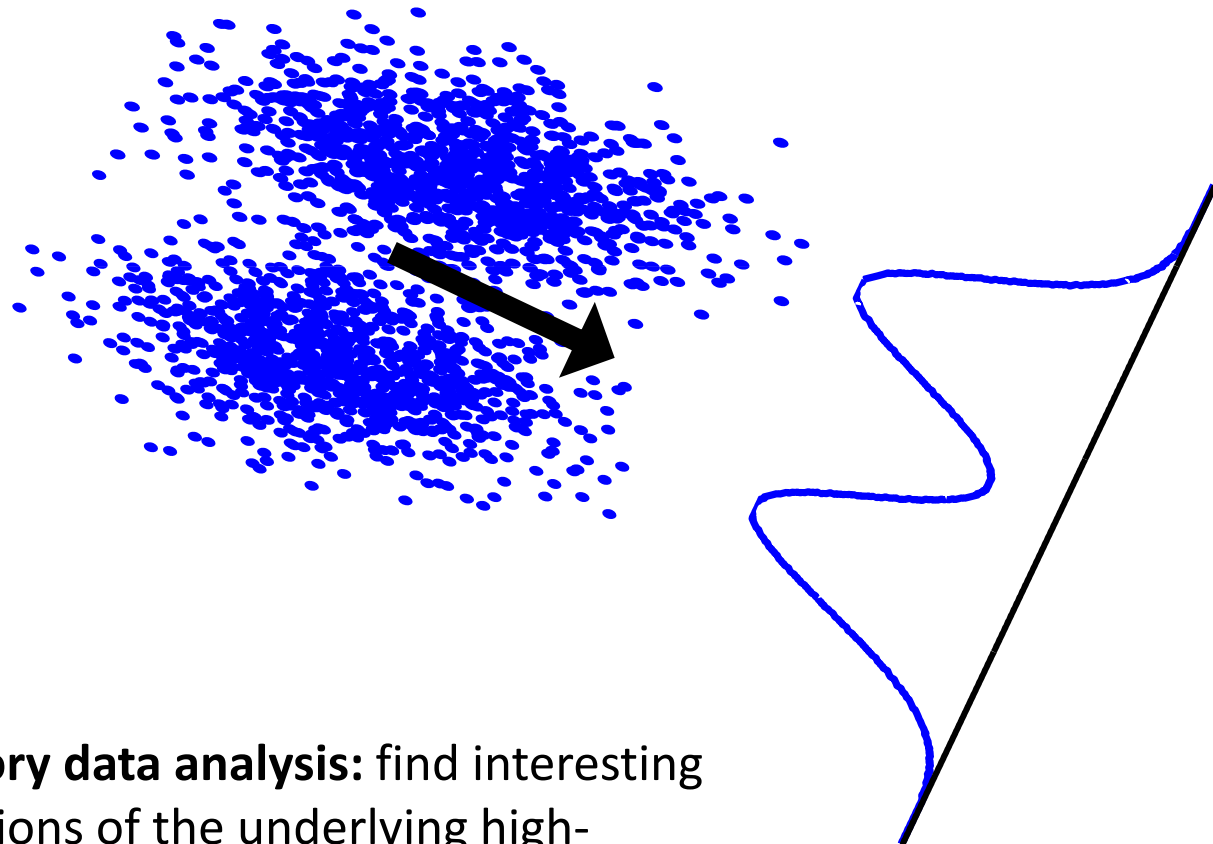
- Lets try a few more projections



Observation: a *typical* projection essentially looks like a Gaussian distribution!

Detour: Projection Pursuit (PP)

- Of course, there are specific directions that give a wealth of information about the underlying density.



PP goal for exploratory data analysis: find interesting (informative) projections of the underlying high-dimensional distribution.

Detour: PP for Density Estimation (PPDE)

- One can use the PP technique to develop **non-parametric density estimators** for the underlying distribution [FSS84].

How?

- Say we have an initial density estimate p_0
- We iteratively improve the estimate the by choosing a direction θ , and picking a univariate function f that best fits the (projected) data.

$$p_m(x) = p_0(x) \prod_{i=1}^m f_i(\theta_i \cdot x)$$

advantage over kernel density estimators: each time only estimating a 1D density, so the effects of the curse of dimensionality is **mitigated**

RBM: structure learning with PPDE [FH91]

- Recall: probability assigned to a (real-valued) state x by a RBM

$$P[x|W] \propto e^{-\frac{1}{2}\|x\|^2} \prod_{i=1}^m (1 + e^{-x \cdot W_{:i}})$$

This closely mimics the functional form of a PPDE!

$$p_m(x) = p_0(x) \prod_{i=1}^m f_i(\theta_i \cdot x)$$

- Thus, there is a natural iterative algorithm to estimate the number of hidden units in an RBM.

Each iteration corresponds to **adding a hidden unit** and estimating the parameters $W_{:i}$ using the input samples by an EM type procedure.

Repeat till no significant improvement in likelihood.

Talk Outline

- Universality [FH91].
- Learning the structure of RBM [FH91].
- **Hardness of approximate inference [LS10].**

RBM: Hardness of Inference [LS10]

- Recall the probability assigned to a particular state x

$$P[x|W] = \frac{1}{Z} \sum_{h \in \{0,1\}^m} e^{x^\top W h} = \frac{1}{Z} \prod_{i=1}^m (1 + e^{-x \cdot W_{:i}})$$

where, the normalization $Z = \sum_{x \in \{-1,1\}^n, h \in \{0,1\}^m} e^{x^\top W h}$

Theorem: Given some x , W and approximation parameter $c > 1$, define $p = P[x|W]$.

If $P \neq NP$ then, returning a value \hat{p} such that $\frac{1}{c} \cdot p \leq \hat{p} \leq c \cdot p$ is **hard**, even when $c = e^{Kn}$ (where K is a fixed constant).

RBM: Hardness of Inference [LS10]

It is **equivalent** to show that approximating the partition function Z to the same resolution is hard.

To this end, we shall use a recent result [AN04]

Lemma 1: If $P \neq NP$ then, exists $\epsilon > 0$ such that approximating

$$\|W\|_\epsilon := \max_{x \in \{-1, +1\}^n, h \in \{0, 1\}^n} x^T W h$$

to within factor $1 + \epsilon$ is hard.

As a consequence, we have the following:

RBM: Hardness of Inference [LS10]

Lemma 2: If $P \neq NP$ then, exists $\alpha > 0$ such solving the following promise problem is hard. Let $f(n) \in \omega(n)$

Input: An $n \times n$ matrix W such that $\max_{i,j} |W_{ij}| \leq f(n)$ and either
(i) $\|W\|_c > f(n)$; or (ii) $\|W\|_c \leq (1 - \alpha)f(n)$

Output: Answer whether (i) or (ii) holds.

Proof sketch: Suppose (for contradiction), for every $\alpha > 0$, ALG_α efficiently solves the promise problem. Then, we can efficiently approximate $\|W\|_c$ for all $\epsilon > 0$ (contradicting Lemma 1).

How? Since $\max_{i,j} |W_{ij}| \leq \|W\|_c \leq n^2 \max_{i,j} |W_{ij}|$, maintain an interval $[l, b]$ of possible values for $\|W\|_c$ and by **repeatedly calling** ALG_α do a **binary search** type pruning of the interval. ■

RBM: Hardness of Inference [LS10]

Lemma 3: If $P \neq NP$ then, exists $\alpha > 0$ such solving the following promise problem is hard. Let $f(n) \in \omega(n)$

Input: An $n \times n$ matrix W such that $\max_{i,j} |W_{ij}| \leq f(n)$ and either

- (i) $\sum_{x,h} e^{x^T W h} > e^{f(n)}$; or
- (ii) $\sum_{x,h} e^{x^T W h} \leq 4^n e^{(1-\alpha)f(n)}$

Output: Answer whether (i) or (ii) holds.

Proof sketch: By previous lemma, we know that $\max_{i,j} |W_{ij}| \leq f(n)$ and either (a) $\max_{x,h} \{x^T W h\} > f(n)$; or (b) $\max_{x,h} \{x^T W h\} \leq (1 - \alpha)f(n)$

It is hard to determine (a) or (b).

Observe: (a) \Rightarrow (i) and (b) \Rightarrow (ii). So, for sufficiently large n , efficiently solving this problem, efficiently solves for the previous problem. ■

RBM: Hardness of Inference [LS10]

Theorem 4: Let $f(n) \in \omega(n)$ and W a matrix satisfying $\max_{i,j} |W_{ij}| \leq f(n)$. If $P \neq NP$ then, exists $\epsilon > 0$ such that approximating $\sum_{x,h} e^{x^T W h}$ to a multiplicative factor of $e^{\epsilon f(n)}$ is hard.

Proof sketch: Let $\alpha > 0$ be from previous lemma.

Set $U = e^{f(n)}$, $L = 4^n e^{(1-\alpha)f(n)}$

If an algorithm can approximate $Z := \sum_{x,h} e^{x^T W h}$ within factor $\sqrt{\frac{U}{L}}$

It can also distinguish $Z \geq U$ from $Z < L$

Note: an approx. better than this contradicts the previous lemma.

This gives us the approximation: $\sqrt{\frac{U}{L}} = e^{(\alpha/2)f(n) - (n/2) \ln 4}$ ■

Conclusion

- RBMs are **simple yet powerful** networks that can encode *any* distribution over $\{-1,1\}^n$!
- There is a simple PPDE-type algorithm that can estimate **the right number** of hidden units for the particular dataset.
- Approximate inference in RBMs is **hard**.

Questions/discussion

References

[AN04] Alon and Naor. Approximating the cut-norm via Grothendieck's inequality. *STOC* 2004.

[FH91] Freund and Haussler. Unsupervised learning of distributions on binary vectors using two layer networks. *NIPS* 1991.

[FSS84] Friedman, Stuetzle and Schroeder. Projection pursuit density estimation. *J. Amer. Stat. Assoc.* 1984.

[FT74] Friedman and Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. on Computers*, 1974.

[LS10] Long and Servedio. Restricted Boltzmann Machines are hard to approximately evaluate or simulate. *ICML* 2010.