limitations of autoregressive models and their alternatives

Chu-Cheng Lin#, Aaron Jaech*, Xin Li#, Matt Gormley*, Jason Eisner#

#Johns Hopkins University
*Facebook AI
* Carnegie Mellon University
This talk
This talk

autoregressive models

\[ p(\text{"Roses are red"}) = 0.1 \quad 0.3 \quad 0.05 \]
This talk

Energy-based models (EBMs)

Roses are red

goodness('Roses are red') = 100

autoregressive models

p('Roses are red') = 0.1 0.3 0.05
This talk

energy-based models (EBMs) \textcolor{red}{\Rightarrow} autoregressive models

goodness(‘Roses are red’) = \textcolor{red}{100}
p(‘Roses are red’) = \begin{align*}
0.1 \\
0.3 \\
0.05
\end{align*}
This talk

goodness\('\text{Roses are red}'\) = 100

\[ p('\text{Roses are red}') = \begin{pmatrix} 0.1 \\ 0.3 \\ 0.05 \end{pmatrix} \]

energy-based models (EBMs)

autoregressive models
Autoregressive models are not as expressive as other model families, energy-based models in particular.

And having more parameters helps little!

Model families that are more expressive than autoregressive models made their own trade-offs
Outline

• Autoregressive models are not as expressive as other model families, energy-based models in particular.

• And having more parameters helps little!

• Model families that are more expressive than autoregressive models made their own trade-offs
Many NLP tasks are about scoring strings

- Language modeling
  - Good:
    Roses are red
  - Maybe:
    Roses are nosy
  - Bad:
    Roses queen sierra

- Machine translation
  - Good:
    Roses are red -> Las rosas son rojas
  - Bad:
    Roses are red -> Las rosas son rojos
Many NLP tasks are about scoring strings we want to measure their goodness quantitatively with an NN

- **Good:**
  Roses are red

- **Maybe:**
  Roses are nosy

- **Bad:**
  Roses queen sierra

![Diagram showing a feedforward layer with BERT input and output connections.](image-url)
Many NLP tasks are about scoring strings we want to measure their goodness quantitatively with an NN

- **Good:**
  Roses are red

- **Maybe:**
  Roses are nosy

- **Bad:**
  Roses queen sierra
Many NLP tasks are about scoring strings
we want to measure their goodness quantitatively with an NN

- **Good:**
  Roses are red

- **Maybe:**
  Roses are nosy

- **Bad:**
  Roses queen sierra

FEEDFORWARD LAYER

BERT

Roses queen sierra

0.01
Many NLP tasks are about scoring strings, we want to measure their goodness quantitatively with an NN.

- \(\text{goodness(“Roses are red”)} = 1000\)
- \(\text{goodness(“Roses are nosy”)} = 5\)
- \(\text{goodness(“Roses queen sierra”)} = 0.01\)
- support of goodness:
  - set of strings whose goodness > 0

Known as energy-based models:

\[\text{goodness}(x)\]

FEEDFORWARD LAYER

\[x = x_1 \ x_2 \ ... \ x_n\]
Many NLP tasks are about scoring strings
we want to measure their goodness quantitatively with an NN

- goodness(“Roses are red”) = 1000
- goodness(“Roses are nosy”) = 5
- goodness(“Roses queen sierra”) = 0.01
- support of goodness:
  - set of strings whose goodness > 0

known as energy-based models

\[ \text{goodness}(x) \]

evaluates in \( O(\text{poly}(n)) \)

**FEEDFORWARD LAYER**

\[ x = x_1 \ x_2 \ ... \ x_n \]
Many NLP tasks are about scoring strings
we want to measure their goodness on a scale between 0 and 1

• $Z = \text{goodness(“Roses are red”)}$
  + $\text{goodness(“Roses are nosy”)}$
  + $\text{goodness(“Roses queen sierra”)}$
  + ...

• intractable!

<table>
<thead>
<tr>
<th></th>
<th>Roses are red</th>
<th>Roses are nosy</th>
<th>Roses queen</th>
<th>Las rosas son rojos</th>
<th>...</th>
<th>SUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>5</td>
<td>0.01</td>
<td>0.00001</td>
<td>1000+5+0.01+0.00001</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Autoregressive parametrization

- goodness("Roses are red") =...
Autoregressive parametrization

- goodness(“Roses are red”) = \( p(“Roses”) \) ...
  = 0.1...
Autoregressive parametrization

- goodness(“Roses are red”) = p(“Roses”) * p(“are” | “Roses”) ... = 0.1 * 0.3 ...

<table>
<thead>
<tr>
<th></th>
<th>red</th>
<th>Roses</th>
<th>are</th>
<th>sierra</th>
<th>...</th>
<th>SUM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.001</td>
<td>0.002</td>
<td>0.3</td>
<td>0.0002</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

<softmax>

<s> Roses
Autoregressive parametrization

• goodness("Roses are red") =
  \( p("Roses") \) * \( p("are" | "Roses") \) * \( p("red" | "Roses are") \)
  = 0.1 * 0.3 * 0.05

<table>
<thead>
<tr>
<th>red</th>
<th>Roses</th>
<th>are</th>
<th>sierra</th>
<th>...</th>
<th>SUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>0.02</td>
<td>0.0007</td>
<td>0.0006</td>
<td>...</td>
<td>1</td>
</tr>
</tbody>
</table>

<s> Roses are
Autoregressive parametrization

- goodness("Roses are red") = 0.0015
- goodness("Roses are nosy") = 0.0000076
- goodness("Roses queen sierra") = 0.000000015

\[
p(x_t | x_{<t}) =
\]

| \(x_t\)   | red | Roses | are | ...
|----------|-----|-------|-----|-----

TRANSFORMER

SOFTMAX

\(<s> \ldots X_{t-1}\)
Autoregressive parametrization

- goodness(“Roses are red”) = 0.0015
- goodness(“Roses are nosy”) = 0.0000076
- goodness(“Roses queen sierra”) = 0.000000015

\[
p(x_t | x_{<t}) = \text{softmax} \]

Evaluates in \(O(\text{poly}(t))\)
Autoregressive parametrization

- Autoregressive models guarantee $Z = 1$
- Goodness("Roses are red") = $0.0015/Z = 0.0015$
- Goodness("Roses are nosy") = $0.0000076/Z = 0.0000076$
- Goodness("Roses queen sierra") = $0.000000015/Z = 0.000000015$

$$x_t = \text{red} \quad \text{Roses} \quad \text{are} \quad \ldots$$

$$p(x_t | x_{<t}) =$$

\[<s> \quad \ldots \quad x_{t-1}\]
EBMs vs autoregressive models

\[ x = x_1 \ x_2 \ ... \ x_n \]

\[ \text{goodness}(x) \]

FEEDFORWARD LAYER

SOFTMAX

TRANSFORMER

\[
\begin{array}{c|cccc}
\text{x}_t= & \text{red} & \text{Roses} & \text{are} & \text{...} \\
\text{p}(\text{x}_t|\text{x}_{<t})= & \text{...} & \text{...} & \text{...} \\
\end{array}
\]
EBMs are more powerful!

\[
goodness(x) = x_1 x_2 \ldots x_n
\]

\[
x_t = \text{red} \quad \text{Roses} \quad \text{are} \quad \ldots
\]

\[
p(x_t | x_{<t}) =
\]

FEEDFORWARD LAYER

EBMs

SOFTMAX

autoregressive models
This work

lookup models \(\leadsto\) (autoregressive) latent variable models \(\leadsto\) EBMs \(\leadsto\) autoregressive models
Why EBM \( \supseteq \) autoregressive models?

\[
x = x_1 \ x_2 \ ... \ x_n
\]

\[
goodness(x)
\]

\[
x_t = \text{red} \ \text{Roses} \ \text{are} \ ... \\
p(x_t | x_{<t}) = ...
\]
Why EBMs $\nsubseteq$ autoregressive models?

- formula: $(A_1 \text{ or not } A_2) \text{ and } (A_3)$
- assignment: 101

$$x = \text{formula } \# \text{ assignment}$$
Why EBMs \( \supseteq \) autoregressive models?

- **formula:**
  \((A_1 \text{ or not } A_2) \text{ and } (A_3)\)

- **assignment:**
  101

- **goodness**(**x**)?
  - > 0 if assignment satisfies formula
  - =0 otherwise

**x = formula # assignment**
Why EBMs ≠ autoregressive models?

- formula: $(A_1 \text{ or not } A_2) \text{ and } (A_3)$
- assignment: 101
- goodness($x$)
  - $> 0$ if assignment satisfies formula
  - $= 0$ otherwise
- goodness($x$) can be constructed as an RNN with size $O(|x|^3)$

$x =$ formula # assignment

goodness($x$)

FEEDFORWARD LAYER
Why EBMs $\supseteq$ autoregressive models?

• Now let’s look at autoregressive models.

• Can we implement goodness($x$) using an autoregressive model?

<table>
<thead>
<tr>
<th>$x_t$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(x_t</td>
<td>x_{&lt;t})$</td>
<td>?</td>
</tr>
</tbody>
</table>

TRANSFORMER

SOFTMAX

<s> ... $x_{t-1}$
Why EBMs $\supseteq$ autoregressive models?

- Computing the first token right after `<s> formula #` is as hard as determining if `formula` is satisfiable...

In contrast, EBMs need only to check a single assignment.
Why EBMs $\not\supset$ autoregressive models?

- Computing the first token right after `<s> formula #` is as hard as determining if formula is satisfiable...
  - which is NP-complete!
- Thus, **no** polynomial-time autoregressive model can model such distributions if $P \neq NP.$
Actually it got worse

• There are distributions that can be captured by EBMs. But no autoregressive model can:
  • capture those distributions exactly (Theorem 1)
  • approximate well enough to get the same ranking of strings (Theorem 2)
  • approximate within any multiplicative factor (Theorem 4)
• Why should we care if we only need to model finite datasets?
  • in other words, we can always make the model larger to handle longer sequences (if smaller models don’t work)...right?
• If the model sizes only grow polynomially in sequence length, they belong in the P/poly class.

• It is widely believed NP ∉ P/poly.

• So the models must grow superpolynomially larger and/or run superpolynomially longer in sequence length, to model longer problems (since they are NP-hard).

• otherwise, the model simply won’t fit even with access to an oracle in training!
What if I am not interested in Boolean SAT problems?

- In general, autoregressive models cannot capture distributions over strings of the form problem#solution, where a problem is computationally hard to solve.
  - EBMs can capture such distributions
- Some CL/NLP problems are indeed computationally hard:
  - Parsing of many syntactic/semantic formalisms (e.g. AMR)
  - Propositional logic (NLI)
  - Optimality Theory
- Important linguistic regularities cannot be captured by autoregressive models!
- We use propositional logic generating a Star Wars movie script as an example.
A long time ago in a galaxy far, far away…. The Rebels fought against the evil Galactic Empire, and eventually won. The story started with Luke….
A long time ago in a galaxy far, far away....
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke....
A long time ago in a galaxy far, far away....
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke....
A long time ago in a galaxy far, far away….
The Rebels fought against the evil Galactic Empire, and eventually won.
The story started with Luke….
A long time ago in a galaxy far, far away….
The Rebels fought against the evil Galactic Empire, and eventually won.
The story started with Luke….
A long time ago in a galaxy far, far away…. The Rebels fought against the evil Galactic Empire, and eventually won. The story started with Luke….

A long time ago in a galaxy far, far away…
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke….

Luke and Vader enjoyed their friendship.

‘friend!’
A long time ago in a galaxy far, far away....
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke....

Luke, I'm your...

'friend!'

Luke and Vader enjoyed their friendship.

Vader turned Luke to the Dark Side.

some plots
A long time ago in a galaxy far, far away....
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke....

Luke, I’m your...

‘friend!’

Luke and Vader enjoyed their friendship.

Vader turned Luke to the Dark Side.

some plots
A long time ago in a galaxy far, far away....
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke....

Luke and Vader enjoyed their friendship.

'friend!'

Luke, I'm your...

Vader turned Luke to the Dark Side.

The Rebels lost! (violates what we said at the beginning)
A long time ago in a galaxy far, far away.... The Rebels fought against the evil Galactic Empire, and eventually won. The story started with Luke....

Luke, I’m your...
A long time ago in a galaxy far, far away....
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke....
A long time ago in a galaxy far, far away....
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke....

The Rebels won!

The authors used lookahead to choose this word
A long time ago in a galaxy far, far away....
The Rebels fought against the evil Galactic Empire, and eventually won.
The story started with Luke....
can the autoregressive model learn that these conditional probabilities are small?

p('friend' | ...Luke, I'm your) = 0

A long time ago in a galaxy far, far away....
The Rebels fought against the evil Galactic Empire, and eventually won.
The story started with Luke....
A long time ago in a galaxy far, far away....
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke....

\[ p('friend' \mid \ldots\text{Luke, I'm your}) = 0 \]

can the autoregressive model learn that these conditional probabilities are small?

... turns out we can’t if \( P \neq NP \) (Theorem 4)

Luke, I’m your...

‘father!’

‘friend!’

some plots
A long time ago in a galaxy far, far away.... The Rebels fought against the evil Galactic Empire, and eventually won. The story started with Luke....

Some plots:

Luke, I'm your...

Luke and Vader enjoyed their friendship.

Vader turned Luke to the Dark Side.

The Rebels lost! (violates what we said at the beginning)

No autoregressive model can guarantee $p(\text{upper arc}) < p(\text{lower arc})$! (Theorem 3)

The Rebels won!

Luke and Vader enjoyed their friendship.

'friend!'

'father!'

nooooooo
brief summary so far

• Autoregressive models cannot even guarantee that its generation is consistent (under propositional logic)!
  • This is really bad because checking their (in)consistency is indeed easy.

• Speaking very loosely, autoregressive models cannot tell between a surprising plot twist and an inconsistent continuation.
Outline

- Autoregressive models are not as expressive as other model families, energy-based models in particular.
  - And having more parameters helps little!
- Model families that are more expressive than autoregressive models made their own trade-offs.
Comparison of model families

<table>
<thead>
<tr>
<th></th>
<th>Compact parameters?</th>
<th>Efficient scoring?</th>
<th>Efficient sampling?</th>
<th>Support can be…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autoregressive models</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>Some but not all languages in P</td>
</tr>
</tbody>
</table>
Comparison of model families

<table>
<thead>
<tr>
<th></th>
<th>Compact parameters?</th>
<th>Efficient scoring?</th>
<th>Efficient sampling?</th>
<th>Support can be...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autoregressive models</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>Some but not all languages in P</td>
</tr>
<tr>
<td>Energy-based models</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Comparison of model families

<table>
<thead>
<tr>
<th></th>
<th>Compact parameters?</th>
<th>Efficient scoring?</th>
<th>Efficient sampling?</th>
<th>Support can be…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autoregressive</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>Some but not all languages in P</td>
</tr>
<tr>
<td>models</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Energy-based</td>
<td>✔️</td>
<td>✔️</td>
<td>✘</td>
<td>no efficient factorization</td>
</tr>
<tr>
<td>models</td>
<td></td>
<td></td>
<td></td>
<td>All languages in P</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Comparison of model families

<table>
<thead>
<tr>
<th></th>
<th>Compact parameters?</th>
<th>Efficient scoring?</th>
<th>Efficient sampling?</th>
<th>Support can be…</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Autoregressive models</strong></td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>Some but not all languages in P</td>
</tr>
<tr>
<td><strong>Energy-based models</strong></td>
<td>✔</td>
<td>✔</td>
<td>✘</td>
<td>All languages in P</td>
</tr>
<tr>
<td><strong>Autoregressive latent variable models</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
An autoregressive model may not prefer plot twists to other bland but logically inconsistent continuations… But actually autoregressive models are capable of generating such plot twists. They just need some backstory to help it justify the climax building.

Vader: “Luke, I am your father!”

…

some happy ending
Vader: “Luke, I am your father!”

some happy ending
Anakin went on a trip.

Vader: “Luke, I am your father!”

some happy ending
Anakin went on a trip.

Anakin graduated from school and got married.

Vader: “Luke, I am your father!”

some happy ending
Anakin went on a trip.

Anakin graduated from school and got married.

Anakin’s wife gave birth to two babies — Luke and Leia. Anakin got a new name “Vader” and also a job.

Vader: “Luke, I am your father!”

some happy ending
Anakin went on a trip.

Anakin graduated from school and got married.

Anakin’s wife gave birth to two babies — Luke and Leia. Anakin got a new name “Vader” and also a job.

fatherhood claim can be proven by backstory (lookahead unnecessary)

Vader: “Luke, I am your father!”

some happy ending
Anakin went on a trip.

Anakin graduated from school and got married.

Vader: “Luke, I am your father!”

Anakin’s wife gave birth to two babies — Luke and Leia. Anakin got a new name “Vader” and also a job.

fatherhood claim can be proven by backstory (lookahead unnecessary)

some happy ending

We only want the fourth story (backstory plots are latent variables)
Anakin went on a trip.

Anakin graduated from school and got married.

Vader: “Luke, I am your father!”

Anakin’s wife gave birth to two babies — Luke and Leia. Anakin got a new name “Vader” and also a job.

We only want the fourth story (backstory plots are latent variables)

fatherhood claim can be proven by backstory (lookahead unnecessary)

turns out with the right latent variables, even autoregressive models can do propositional logic. In fact they can model any NP language! (Theorem 6)
## Comparison of model families

<table>
<thead>
<tr>
<th></th>
<th>Compact parameters?</th>
<th>Efficient scoring?</th>
<th>Efficient sampling?</th>
<th>Support can be…</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Autoregressive models</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Some but not all languages in P</td>
</tr>
<tr>
<td><strong>Energy-based models</strong></td>
<td>✓</td>
<td>✓</td>
<td>✘</td>
<td>All languages in P</td>
</tr>
<tr>
<td><strong>Autoregressive latent variable models</strong></td>
<td>✓</td>
<td>✘</td>
<td>✓</td>
<td>All languages in NP</td>
</tr>
</tbody>
</table>

*needs to marginalize*
## Comparison of model families

<table>
<thead>
<tr>
<th></th>
<th>Compact parameters?</th>
<th>Efficient scoring?</th>
<th>Efficient sampling?</th>
<th>Support can be…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autoregressive models</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>Some but not all languages in P</td>
</tr>
<tr>
<td>Energy-based models</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>All languages in P</td>
</tr>
<tr>
<td>Autoregressive latent variable models</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>All languages in NP</td>
</tr>
<tr>
<td>Lookup models</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Lookup models

• if the model size can be unbounded, we can model any finite language!
• Look up factoids in a database
  • there are sub-linear time retrieval methods.
• examples include kNNLM and adaptive semiparametric LMs.
## Comparison of model families

<table>
<thead>
<tr>
<th></th>
<th>Compact parameters?</th>
<th>Efficient scoring?</th>
<th>Efficient sampling?</th>
<th>Support can be…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autoregressive models</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>Some but not all languages in P</td>
</tr>
<tr>
<td>Energy-based models</td>
<td>✔</td>
<td>✔</td>
<td>✘</td>
<td>All languages in P</td>
</tr>
<tr>
<td>Autoregressive latent variable models</td>
<td>✔</td>
<td>✘</td>
<td>✔</td>
<td>All languages in NP</td>
</tr>
<tr>
<td>Lookup models</td>
<td>✘</td>
<td>✔</td>
<td>✔</td>
<td>Anything</td>
</tr>
</tbody>
</table>

- ✔: Supported
- ✘: Not supported

*unbounded size*
Conclusion

• Autoregressive models are inherently limited.

  • Some string distributions have ‘hard’ conditional probabilities, even though the joint (unnormalized) probabilities may be easy to evaluate.

• Alternative model families have their own tradeoffs.