https://eduassistpro.github.io/
Assignment Project Exam Help
Add WeChat edu_assist_pro

Recurrent Neural Networks (RNN)

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Long distance dependency in sequence labeling

https://eduassistpro.github.io/

Assignment Project Exam Help

Limitation of window based feature extraction for linear sequence models can reach a very high accuracy, but are insuffi cases:

Add WeChat edu_assist_pro

Assignment Project

- POS tagging _VBZ pianos
- Named Entit d
  "Pretty soon I don't know what but _I-title is going to happen"

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

- Language modeling: The man who whistles **tunes** pianos
- ...

Simple Recurrent Neural Networks

https://eduassistpro.github.io/
Assignment Project Exam Help
Add WeChat edu_assist_pro

- A simple recurren t. It is defined as:

$$\text{RNN}(\boldsymbol{x}_m, \boldsymbol{h}_{m-1}; \Theta) = g(\quad$$

Assignment Project

where $\Theta \in$ g is a nonlinear transforma wise hyperbolic tangent tanh.

https://eduassistpro.github.io/
Add WeChat edu_assist_pro

- Although each w_m only depends o $\boldsymbol{h}_{m-1}$, but this vector is affected by all previous tokens $w_0, w_1, \cdots, w_{m-1}$. This is crucially different from n-gram language models.
- While in principle this simple RNN can handle long distance dependencies, in practice it is quite inadequate, due to the repeated application of the non-linearity.

Layers in an RNN

https://eduassistpro.github.io/

Assignment Project Exam Help

Add WeChat edu_assist_pro

- How many layers do we have in an RNN?
  - The number of layers in an RNN is not fixed and varies with the lengt
  - Derivat kages such as Torch, MXNet, and TensorFlow
- How many weight matricies do we need fo

Assignment Project

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Parameters of a simple RNN

https://eduassistpro.github.io/

Assignment Project Exam Help

- $\phi_i \in \mathbb{R}^K$, the "input" word vectors (word e
- $\beta_i \in \mathbb{R}^K$, the "output" word vectors;
- $\Theta \in \mathbb{R}^{K \times K}$
- h_0, the

Assignment Project edu_assist_pro

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Each of these parameters can be tuned by forms over the training corpus $L(\boldsymbol{w})$

Backpropagation through time (BPTT)

https://eduassistpro.github.io/
Assignment Project Exam Help
Add WeChat edu_assist_pro
AssignmentProject
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

- Let ℓ_{m+1} represent the negative log-likelihood of word $m+1$,

$$\ell_{m+1} = -\log P(w_{m+1} \mid \ldots)$$

- Since the loss d $\boldsymbol{h}_m$ (not through tiation:

$$\frac{\partial \ell_{m+1}}{\partial \theta_{k,k'}} = \frac{\partial}{m} \quad _{k,k}$$

- What is the derivative of this function with respect to an element in $\boldsymbol{\Theta}$?

Recurrence in gradient computation

https://eduassistpro.github.io/
Assignment Project Exam Help

- Recall the simple RNN with the Elman Unit

$$\boldsymbol{h}_m = g(\boldsymbol{\Theta}\boldsymbol{h}_{m-1} + \boldsymbol{x}_m)$$

Assignment Project Add WeChat edu_assist_pro

- For any element $\theta_{k,k'}$

https://eduassistpro.github.io/

- Applying the chain rule and the product ru

Add WeChat edu_assist_pro

$$\frac{\partial h_{m,k}}{\partial \theta_{k,k'}} = g'(x_{m,k} + \boldsymbol{\theta}_k \cdot \boldsymbol{h}_{m-1}) \quad h_{m-1,k'} + \boldsymbol{\theta}_k \cdot \frac{\partial \boldsymbol{h}_{m-1}}{\partial \theta_{k,k'}}\Big)$$

- This means the derivative of $\frac{\partial \boldsymbol{h}_m}{\partial \theta_{k,k'}}$ depends on $\frac{\partial \boldsymbol{h}_{m-1}}{\partial \theta_{k,k'}}$, which in turn depends on $\frac{\partial \boldsymbol{h}_{m-2}}{\partial \theta_{k,k'}}$, etc., till $\boldsymbol{h}_0$ is reached

Variants of RNNs

https://eduassistpro.github.io/
Assignment Project Exam Help
Add WeChat edu_assist_pro

- An Elman RNN helps us demonstrate RNN's capacity of modeling longer context, and the needs to use the network.

Assignment Project
x_m)
https://eduassistpro.github.io/

- A more common RNN also paramateriz another parameter:

Add WeChat edu_assist_pro

$$\text{RNN}(\boldsymbol{x_m}, \boldsymbol{h}_{m-1}) = g(\boldsymbol{\Theta}^h \boldsymbol{h}_{m-1} + \boldsymbol{\Theta}^x \boldsymbol{x_m})$$

Updating $\boldsymbol{\Theta}^x$ does not require backpropate through time.

Hyperparameters

https://eduassistpro.github.io/

Assignment Project Exam Help

Add WeChat edu_assist_pro

- The optimal size of the word and context vecto K loosely depends on th
- If the training d ered.
- Conversely, a smaller K should b , the model may memorize the traini data a ze.

Assignment Project

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Vanishing or exploding gradients and gated networks

https://eduassistpro.github.io/
Assignment Project Exam Help
Add WeChat edu_assist_pro

- RNNs require repeated applications of the non-linear functions. Backpropagation can lead to van (gradients decay to zero) or exploding gradi increase towards infinity).
- Exploding gr nts (set a maximu
- Vanishing gradients must be addresse el itself. Gated networks such as LSTM (Lo memory) and GRU are popular solutions to this problem.
- You can find a demonstration here: https://cs224d.stanford.edu/notebooks/vanishing_grad_example.html

Gated Recurrent Units (GRUs)

https://eduassistpro.github.io/

Assignment Project Exam Help

Add WeChat edu_assist_pro

- GRU uses a *Reset* gate and an *Update* to control how much information from the previous hidden state t next hidden state.

Assign Project

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Reset gate	$\boldsymbol{r}_{m+1} = \sigma(\boldsymbol{\Theta}^{h \to r} \boldsymbol{h}_m + \boldsymbol{\Theta}^{x \to r} \boldsymbol{x}_{m+1} + \boldsymbol{b}_r)$
update gate	$\boldsymbol{u}_{m+1} = \sigma(\boldsymbol{\Theta}^{h \to u} \boldsymbol{h}_m + \boldsymbol{\Theta}^{x \to u} \boldsymbol{x}_{m+1} + \boldsymbol{b}_u)$
update candidate	$\tilde{\boldsymbol{h}}_{m+1} = \tanh(\boldsymbol{\Theta}^{h} (\boldsymbol{r}_{m+1} \odot \boldsymbol{h}_m) + \boldsymbol{\Theta}^{x \to h} \boldsymbol{x}_{m+1})$
output	$\boldsymbol{h}_{m+1} = \boldsymbol{u}_{m+1} \odot \boldsymbol{h}_m + (1 - \boldsymbol{u}_{m+1}) \tilde{\boldsymbol{h}}_{m+1}$

What is a gate?

https://eduassistpro.github.io/

Assignment Project Exam Help

- "Hard" gate vs "soft" gate:

Assignment Project Add WeChat edu_assist_pro

$$\delta \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0.998 \\ 0 \\ 0 \end{bmatrix}$$

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

- Hard gates are easier to understand conceptually, but soft gates are differentiable (therefore learnable)

GRU gates

https://eduassistpro.github.io/

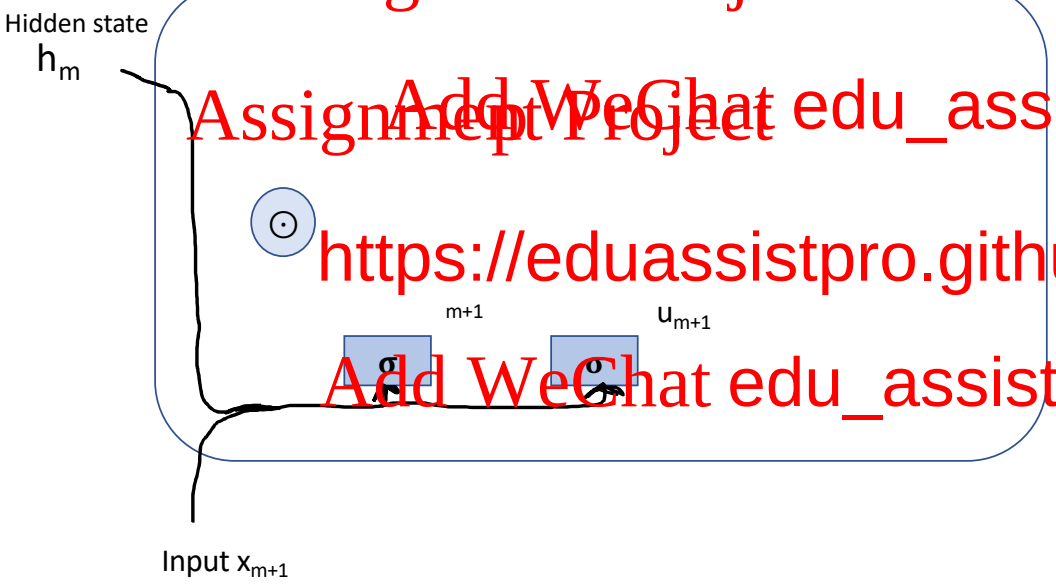
Assignment Project Exam Help

Add WeChat edu_assist_pro

Assignment Project

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

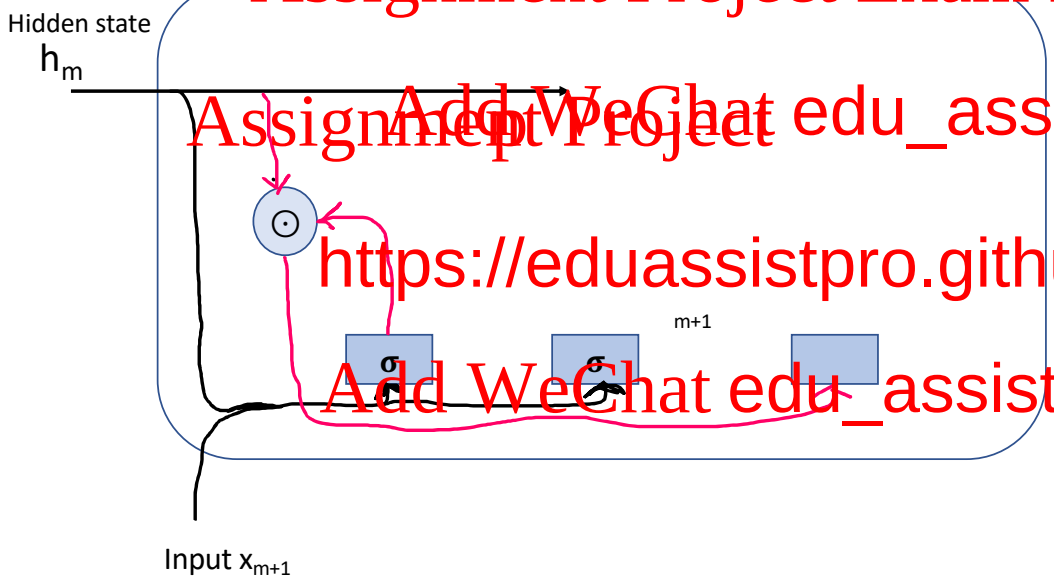


GRU gates

https://eduassistpro.github.io/

Assignment Project Exam Help

Add WeChat edu_assist_pro



Assignment Project

https://eduassistpro.github.io/

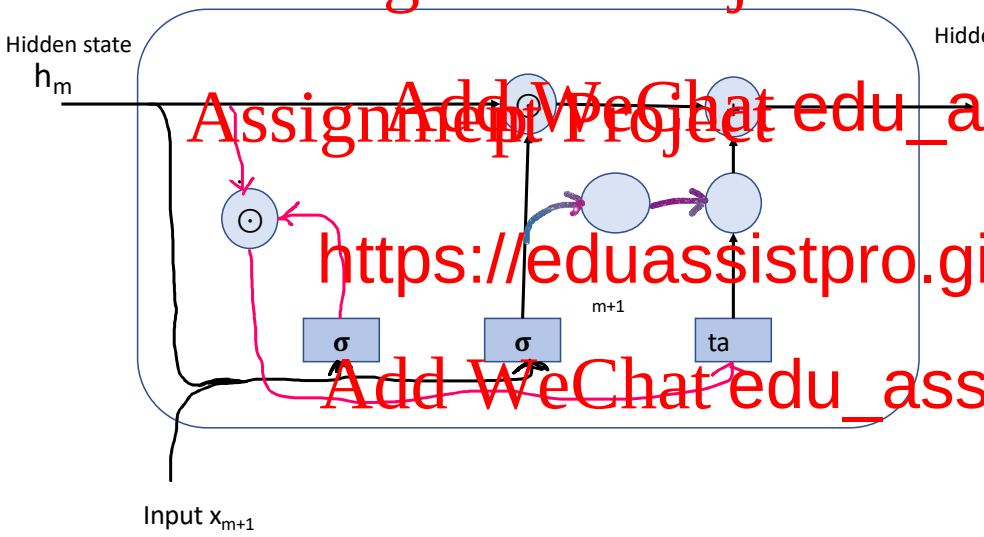
Add WeChat edu_assist_pro

GRU gates

https://eduassistpro.github.io/

Assignment Project Exam Help

Add WeChat edu_assist_pro



Assignment Project

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

GRU: Pytorch implementation

https://eduassistpro.github.io/

Assignment Project Exam Help

Add WeChat edu_assist_pro

Assignment Project

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Long Short Term Memory (LSTM)

https://eduassistpro.github.io/

- LSTM enhances the hidden $\boldsymbol{h}_m$ with a memory cell $\boldsymbol{c}_m$
- This memory cell does not pass through a squashing function (tanh or σ), and can propagate through th distances.

Assignment Project Exam Help
Assignment Project Add WeChat edu_assist_pro
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

forget gate	$\boldsymbol{f}_{m+1} = \sigma(\boldsymbol{\Theta}^{h \to f} \boldsymbol{h}_m + \boldsymbol{\Theta}^{x \to f} \boldsymbol{x}_{m+1} + \boldsymbol{b}_f)$
input gate	$\boldsymbol{i}_{m+1} = \sigma(\boldsymbol{\Theta}^{h \quad i} \quad \boldsymbol{\Theta}^{x \quad i} \boldsymbol{x}_{m+1} + \boldsymbol{b}_i)$
update candidate	$\tilde{\boldsymbol{c}}_{m+1} = \tanh(\boldsymbol{\Theta}^{h \to c} \boldsymbol{h}_m + \boldsymbol{\Theta}^{x \to c} \boldsymbol{x}_{m+1})$
memory cell update	$\boldsymbol{c}_{m+1} = \boldsymbol{f}_{m+1} \odot \boldsymbol{c}_m + \boldsymbol{i}_{m+1} \odot \tilde{\boldsymbol{c}}_{m+1}$
output gate	$\boldsymbol{o}_{m+1} = \sigma(\boldsymbol{\Theta}^{h \to o} \boldsymbol{h}_m + \boldsymbol{\Theta}^{x \to o} \boldsymbol{x}_{m+1} + \boldsymbol{b}_o)$
output	$\boldsymbol{h}_{m+1} = \boldsymbol{o}_{m+1} \odot \tanh(\boldsymbol{c}_{m+1})$

LSTM gates

https://eduassistpro.github.io/

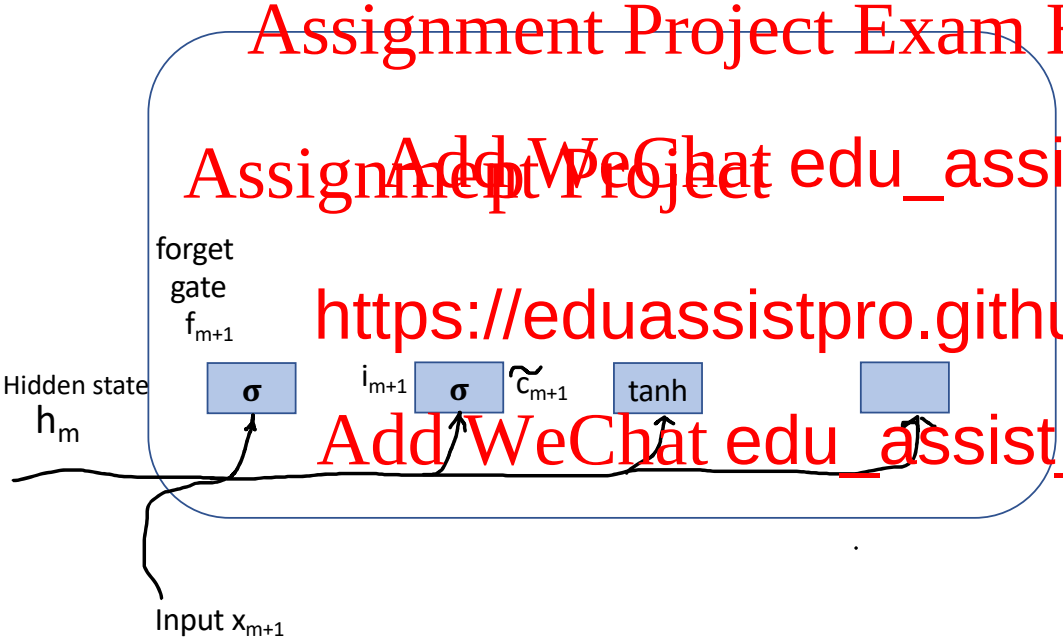
Assignment Project Exam Help

Add WeChat edu_assist_pro

Assignment Project

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

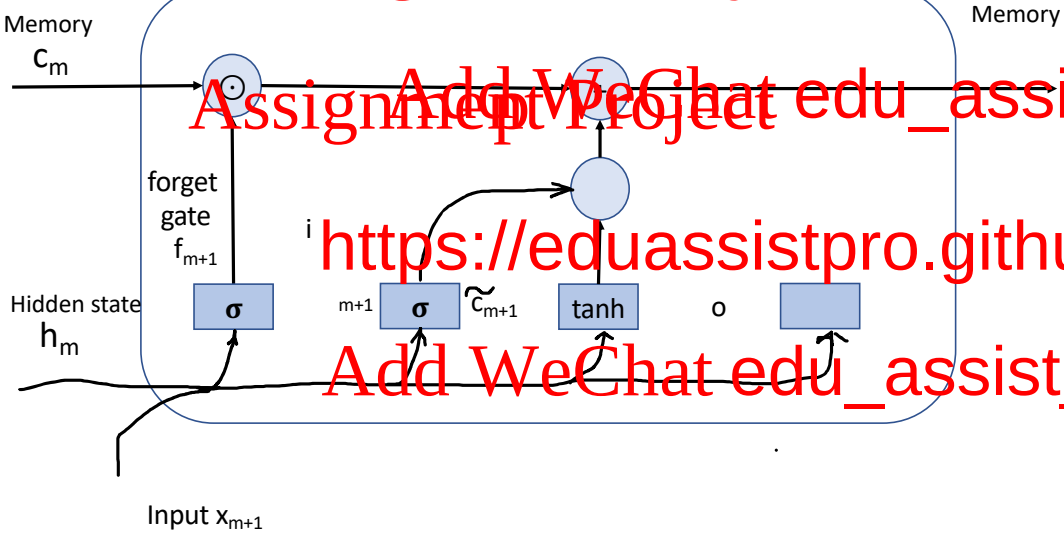


LSTM gates

https://eduassistpro.github.io/

Assignment Project Exam Help

Add WeChat edu_assist_pro



Assignment Project

https://eduassistpro.github.io/

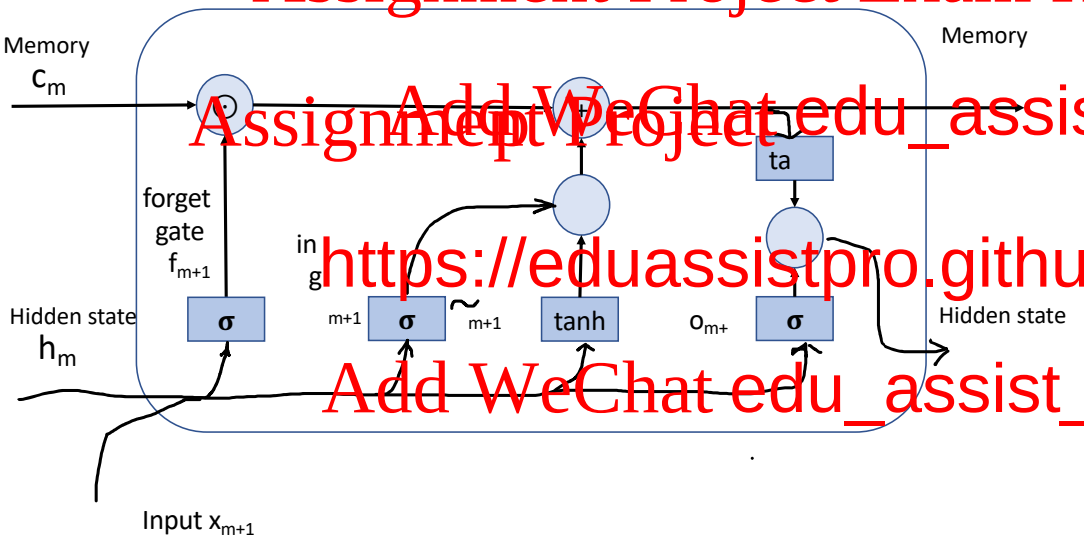
Add WeChat edu_assist_pro

LSTM gates

https://eduassistpro.github.io/

Assignment Project Exam Help

Add WeChat edu_assist_pro



Assignment Project

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

LSTM: Pytorch implementation

https://eduassistpro.github.io/

Assignment Project Exam Help

Add WeChat edu_assist_pro

Assignment Project

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

The successful stories of Recurrent Neural networks in NLP

https://eduassistpro.github.io/

Assignment Project Exam Help

Add WeChat edu_assist_pro

RNN is very versatile and can be used in a variety of different NLP problems:

- Sequence labeling problems: POS tagging, Recognitio
- Sequence to s achine Translation, Machine Reading, Dialo age generation, etc.
- It also serves as an underlying feature extr to many classification problems (e.g., the use of Bi-LSTM RNNs for extracting spans)

Assignment Project

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

RNNs for sequence labeling

- Recall RNN output: one step:

$$\boldsymbol{h}_m = g(\boldsymbol{x}_m, \boldsymbol{h}_{m-1}), m = 1, 2, \ldots, M$$

- Scoring each label with a linear function:

$$\psi_m(y) = \boldsymbol{\theta}_y \cdot \boldsymbol{h}_m$$

- We can also turn this score into a probabilis using SoftMax:

$$P(y|\boldsymbol{w}_{1:m}) = \frac{\exp \psi_m(y)}{\sum_{y' \in \mathcal{Y}} \exp \psi(y')}$$

- This is a classifier that uses only the context from the left.

https://eduassistpro.github.io/
Assignment Project Exam Help
Add WeChat edu_assist_pro
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Bi-directional LSTM (BiLSTM)

https://eduassistpro.github.io/

- Forward LSTM and Backward LSTM

Assignment Project Exam Help

$$\vec{\boldsymbol{h}} = g(\boldsymbol{x}_m, \vec{\boldsymbol{h}}_{m-1}), m$$

Add WeChat edu_assist_pro

$$\overleftarrow{\boldsymbol{h}} = g(\boldsymbol{x}_m, \overleftarrow{\boldsymbol{h}}_{m+1}), m$$

https://eduassistpro.github.io/

- BiLSTM summarizes the surrounding directions, typically better than windowing ngrams both sides of the target word.

Add WeChat edu_assist_pro

- But it doesn't taking into consideration the transitions between tags.

Neural Structure Prediction: LSTM-CRF

Neural sequence labeli
algorithm by defining th

https://eduassistpro.github.io/

Assignment Project Exam Help

$$s_m(y_m, y_{m-1}) = \boldsymbol{\beta} \cdot \boldsymbol{h}_m$$

Add WeChat edu_assist_pro

Assignment Project

https://eduassistpro.github.io/

Add WeChat edu_assist_pro