# Converting equations to LaTeX

Robin Sobczyk, Benjamin Loison, Bastien Lhopitallier

https://github.com/Benjamin-Loison/Equations-images-to-LaTeX

4 février 2022

# Table of contents

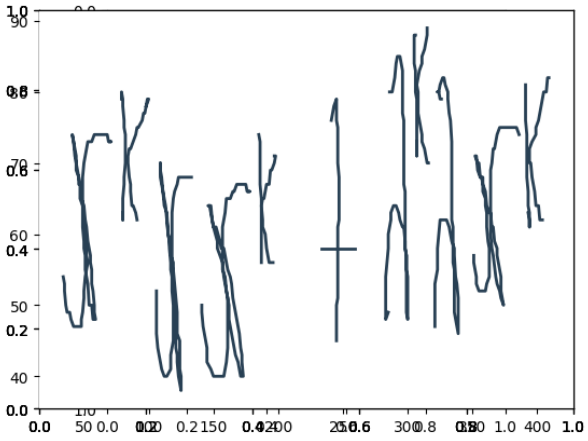
$$\hat{f}(x) = \int_0^1 \left( \frac{\partial f}{\partial x} f'(x) \right) dx$$

$$\downarrow$$

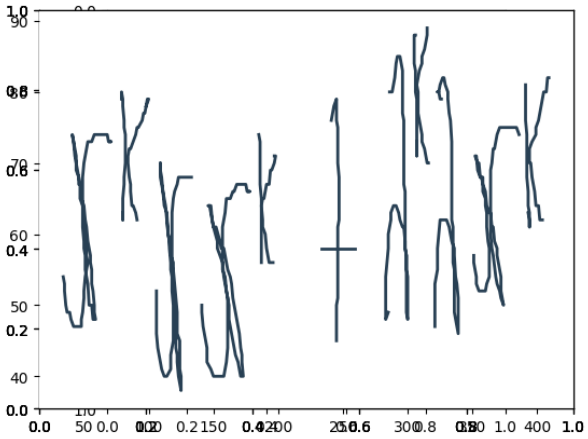
```
\hat { f } ( x ) = \int ^ { 1 } _ { 0 } \left( \dfrac
{ \partial f } { \partial x } f ^ { ’ } ( x ) \right)
d x
```

# Table of contents

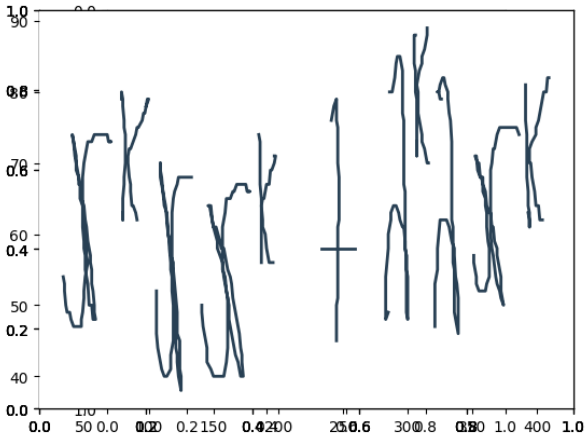
# Handwritten dataset

- INKML to RGB
- 

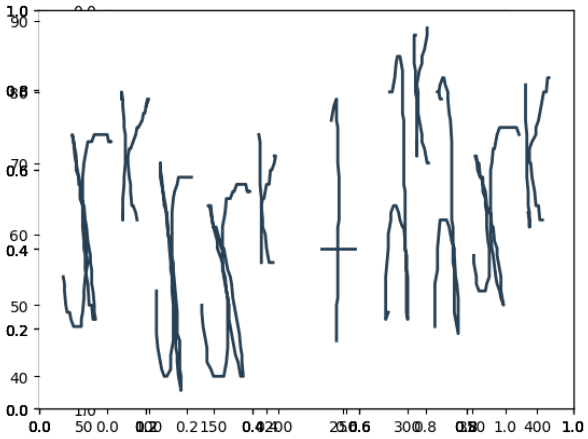
# Handwritten dataset

- INKML to RGB
- 
- Trace : 404 62, ..., 404 62, 402 62, ...

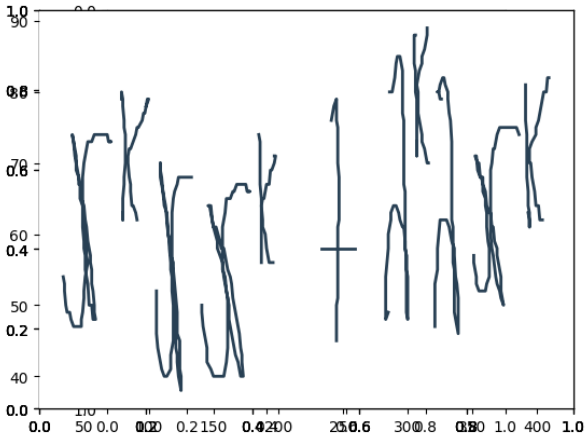
# Handwritten dataset

- INKML to RGB
- 
- Trace : 404 62, ..., 404 62, 402 62, ...
- Remove legends

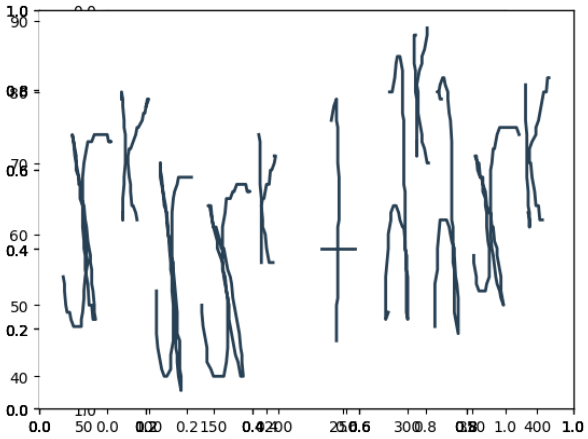
# Handwritten dataset

- INKML to RGB
- 
- Trace : 404 62, ..., 404 62, 402 62, ...
- Remove legends
- Crop images to black to reduce pixels number

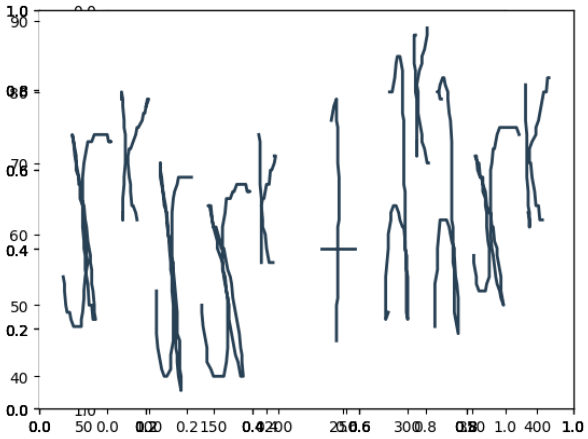
# Handwritten dataset

- INKML to RGB
- 
- Trace : 404 62, ..., 404 62, 402 62, ...
- Remove legends
- Crop images to black to reduce pixels number
- Flip images because upside down

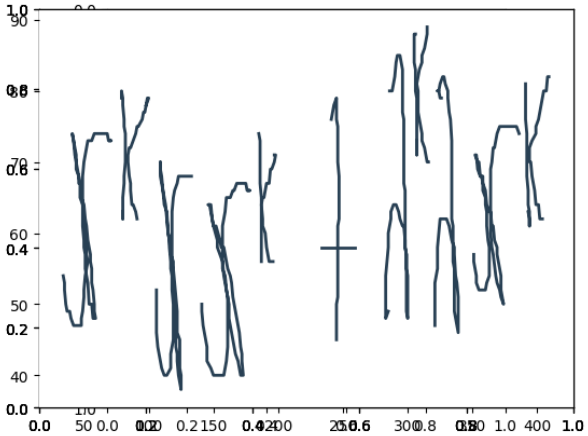
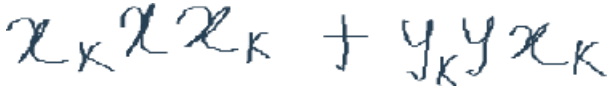
# Handwritten dataset

- INKML to RGB
- 
- Trace : 404 62, ..., 404 62, 402 62, ...
- Remove legends
- Crop images to black to reduce pixels number
- Flip images because upside down
- Resize nearest neighbor

# Handwritten dataset

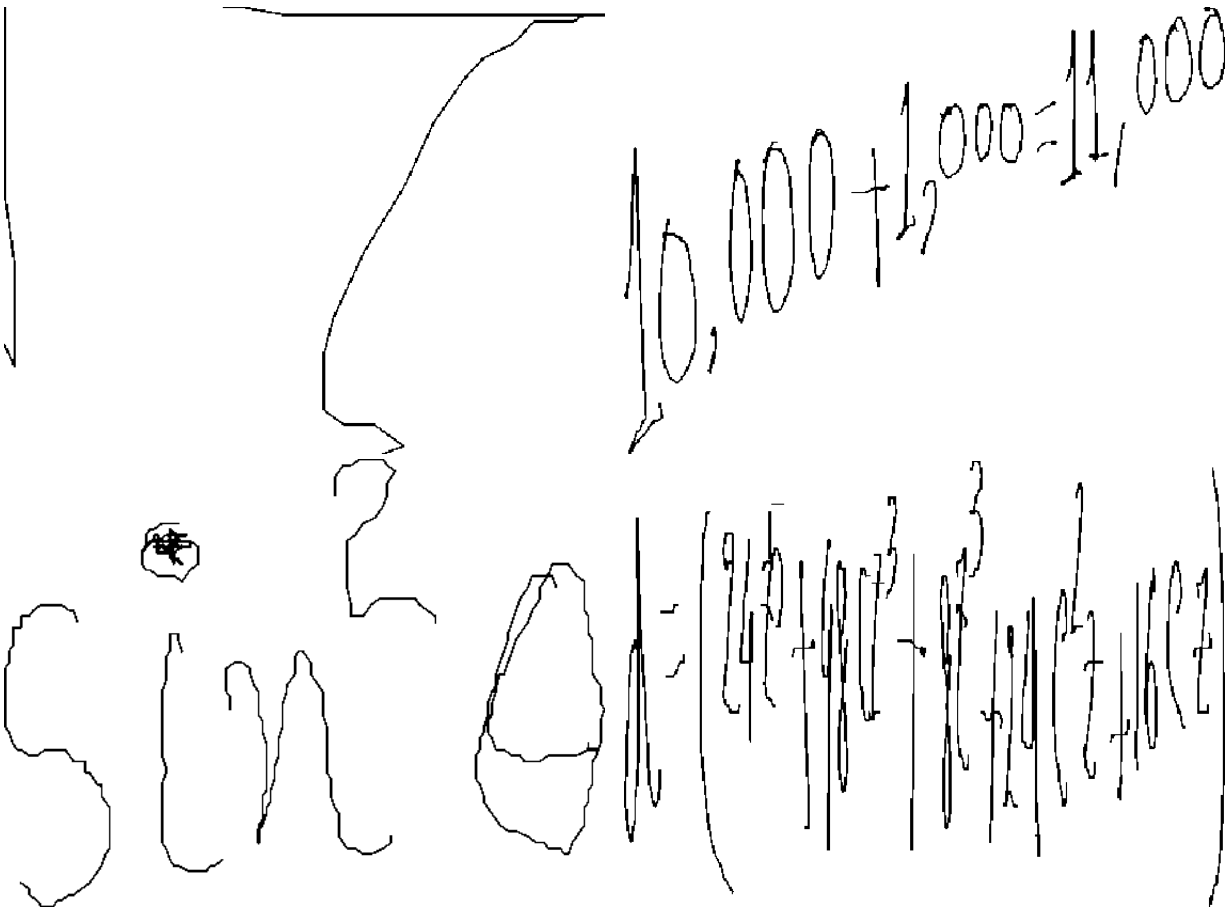
- INKML to RGB
- 
- Trace : 404 62, ..., 404 62, 402 62, ...
- Remove legends
- Crop images to black to reduce pixels number
- Flip images because upside down
- Resize nearest neighbor
- + statistics to get max rectangle taken by black pixels

# Handwritten dataset

- INKML to RGB
- 
- Final result
- 
- Label : $x_k xx_k + y_k yx_k$
- Finally abandonned because don't know how to normalize images

# LaTeX images generated dataset

im2latex-100k



# Compiled LaTeX images dataset

Need some preprocess to transform :

| Transform | Original | Normalized |
|---|---|---|
| SubSup | H^I_1 | H_I^1 |
| ReqBrack | H^I | H^{I} |
| Desugar | H' | H^{\prime} |
| ExpOperators | \sin | \operatorname{sin} |
| InfixPrefix | \over | \frac{}{} |
| MatrixEnv | \matrix | \begin{array}... |
| Drop | \label{} | - |

Table 1: Preprocessing transformations applied to LaTeX abstract syntax tree in normalizing mode. These transformations are mostly safe, although there are some corner cases where they lead to small differences in output.

- `ds ^{2} = (1 - {qcos\theta\over r})`
- ['d', 's', '^', '{', '2', '}', '=', '(', '1', '-', '{', '\frac', '{', 'q', 'c', 'o', 's', '\theta', '}', '{', 'r', '}', '}', ')']
- [0, 11, 31, 9, 5, 32, 8, 28, 23, 33, 6, 5, 18, 5, 34, 35, 36, 31, 37, 8, 5, 38, 8, 8, 24, 1]

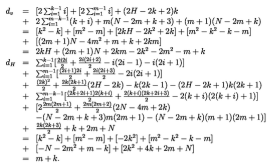
# Compiled LaTeX images dataset

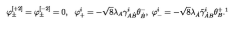
Need some preprocess to transform :

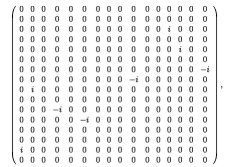
- Removed 5 very unexpected images to reduce black rectangle area used to reduce pixels number
- Likewise total pixels number from 64,000,000,000 to 11,764,000,000, it is a 5.44 factor
- Furthermore these 5 images weren't in train, nor test, nor validation set (10 % of images are in this case)
- Pad images to normalize, using Python but slow so used C++ with multithreading

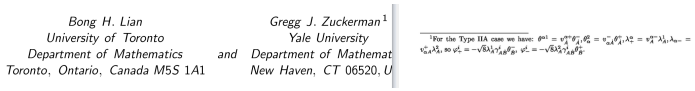
# LaTeX images generated dataset

im2latex-100k







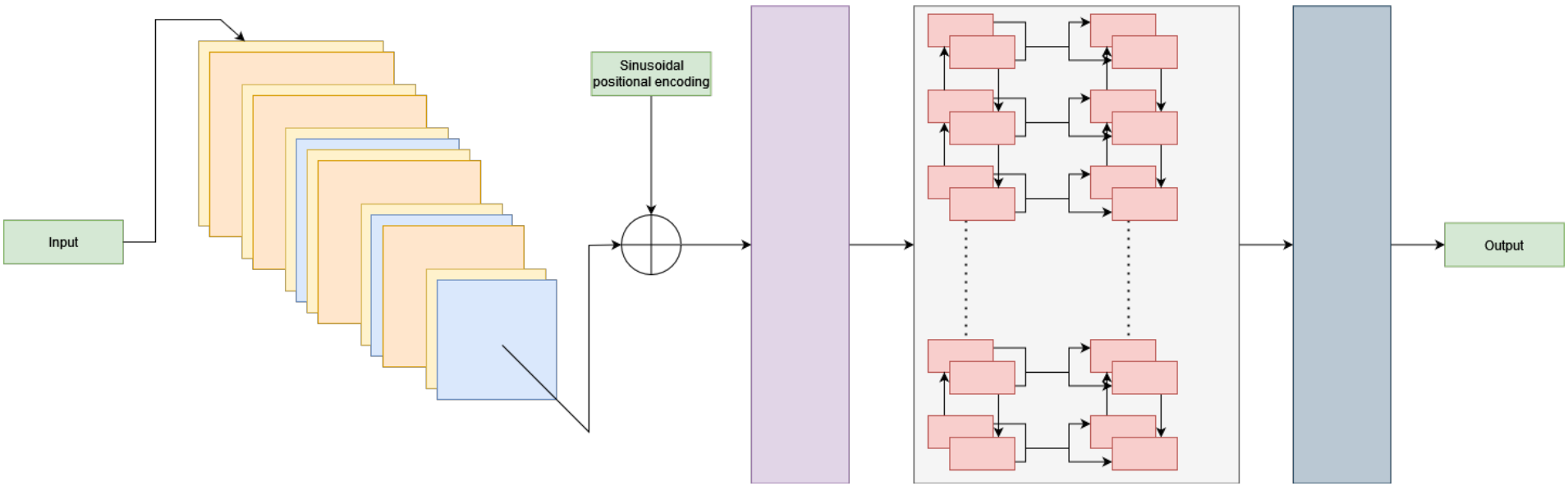


White image and 19x : M a t h T y p e ! Z Z h x 4 7 ! c a a a d a G c b i a H W n W d b a W c b i G H R a q e a O G a g 2 Z a b a a a b u q a b e G a c a a a b i

# Table of contents

# Conduct of the model

## Architecture of the network




| | | |
|---|---|---|
| green : data | yellow : convolution | orange : pooling |
| blue : batch normalization | purple : embedding | light grey : RNN |
| $\oplus$ : element-wise addition | dark grey : Gumbel softmax | red : LSTM |

# Table of contents

After hours of studies and understanding

# Results

After hours of studies and understanding, hours of implementation

# Results

After hours of studies and understanding, hours of implementation
We achieved to get

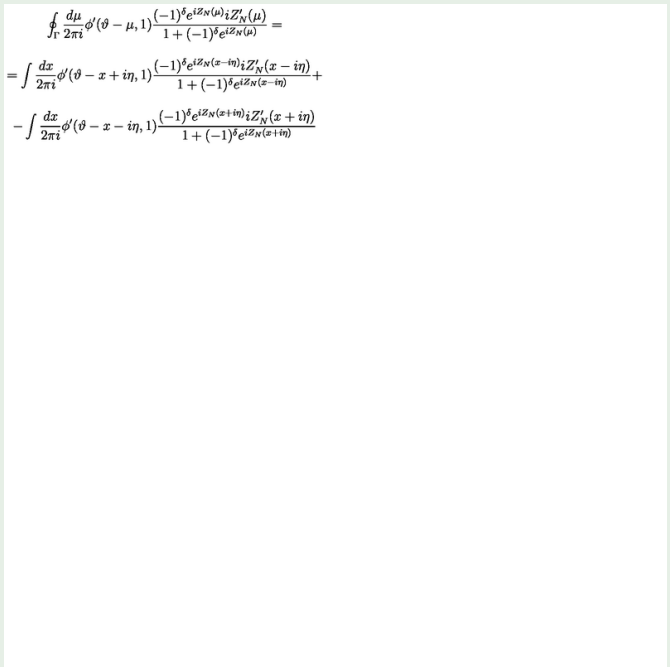
After hours of studies and understanding, hours of implementation
We achieved to get

# No result, it failed miserably

# Flaws in the dataset

## Examples

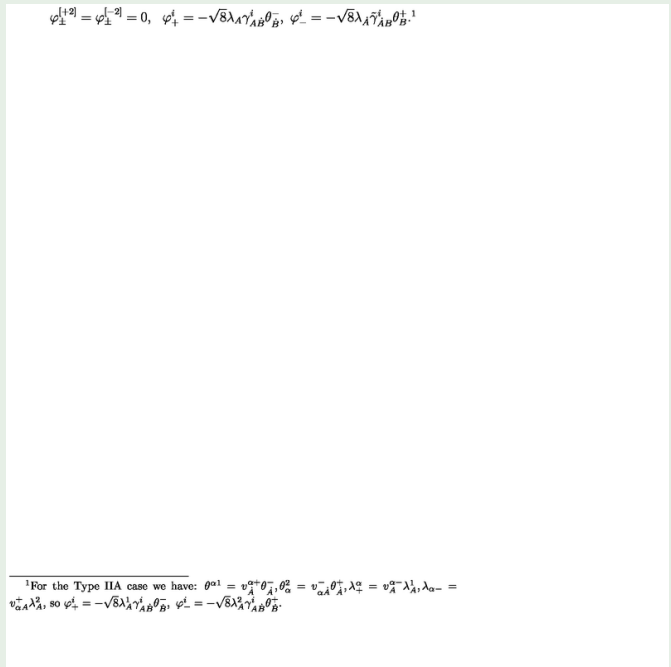
- pictures mostly composed of white pixels (even some that are completely blank)
- huge gaps between parts of the formulas



$$\oint_\Gamma \frac{d\mu}{2\pi i}\phi'(\vartheta-\mu,1)\frac{(-1)^\delta e^{iZ_N(\mu)}iZ'_N(\mu)}{1+(-1)^\delta e^{iZ_N(\mu)}} =$$

$$= \int \frac{dx}{2\pi i}\phi'(\vartheta-x+i\eta,1)\frac{(-1)^\delta e^{iZ_N(x-i\eta)}iZ'_N(x-i\eta)}{1+(-1)^\delta e^{iZ_N(x-i\eta)}} +$$

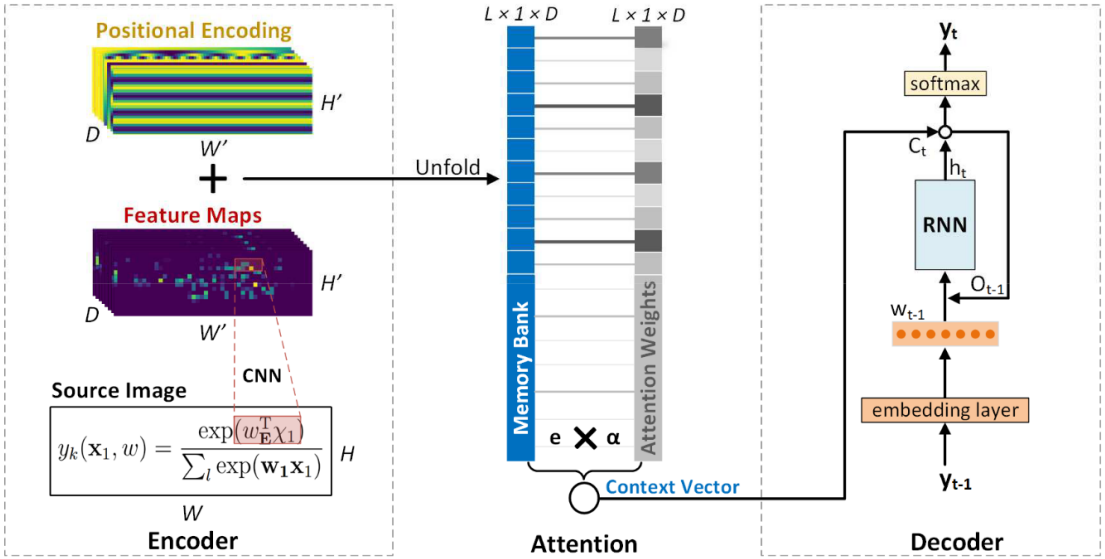
$$- \int \frac{dx}{2\pi i}\phi'(\vartheta-x-i\eta,1)\frac{(-1)^\delta e^{iZ_N(x+i\eta)}iZ'_N(x+i\eta)}{1+(-1)^\delta e^{iZ_N(x+i\eta)}}$$

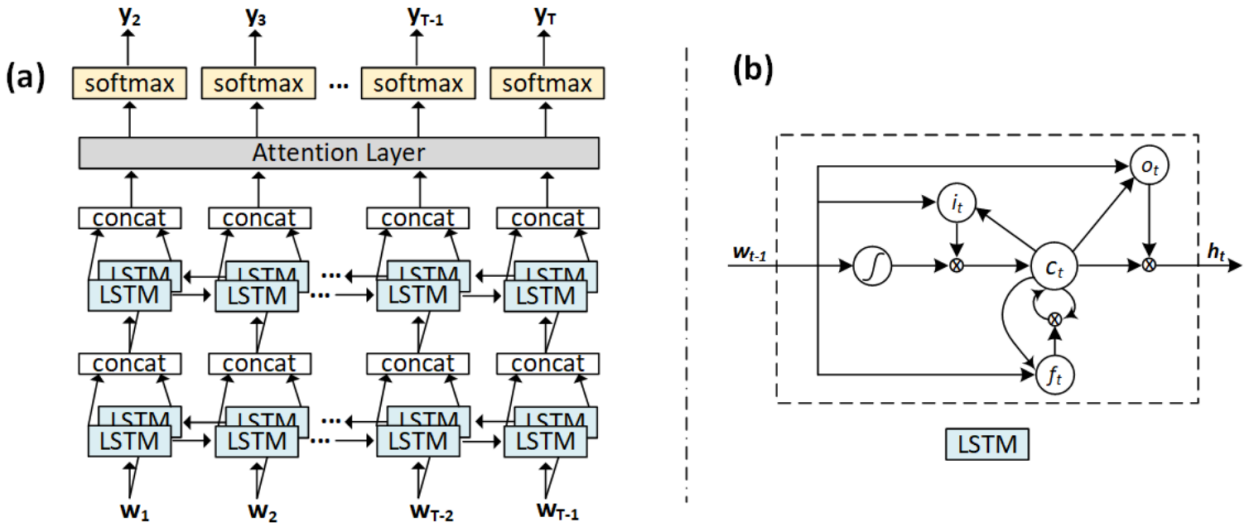


$$\varphi_\perp^{[+2]} = \varphi_\perp^{[-2]} = 0, \quad \varphi_+^i = -\sqrt{8}\lambda_A\gamma^i_{A\dot{B}}\theta^-_{\dot{B}}, \ \varphi_-^i = -\sqrt{8}\lambda_A\tilde{\gamma}^{i\,\dot{A}B}\theta^+_B.^1$$

[1] For the Type IIA case we have: $\theta^{\alpha 1} = v^{\alpha+}_A\theta^-_{\dot{A}}, \theta^2_\alpha = v^{-}_{\alpha\dot{A}}\theta^+_{\dot{A}}, \lambda^\alpha_+ = v^{\alpha-}_A\lambda^1_{\dot{A}}, \lambda_{\alpha-} = v^{+}_{\alpha A}\lambda^2_A$, so $\varphi^i_+ = -\sqrt{8}\lambda^1_A\gamma^i_{A\dot{B}}\theta^-_{\dot{B}}$, $\varphi^i_- = -\sqrt{8}\lambda^2_A\gamma^i_{A\dot{B}}\theta^-_{\dot{B}}$.

# Incoherences in the article



# Sources


[1] Zelun Wang & Jyh-Charn Liu, *Translating Math Formula Images to LaTeX Sequences Using Deep Neural Networks with Sequence-level Training*

[2] Deng Y., Kanervisto A., Ling J. & Rush A. M., *Image-to-Markup Generation with Coarse-to-Fine Attention*

[3] Zhang J., Du J. & Dai L., *A GRU-based Encoder-Decoder Approach with Attention for Online Handwritten Mathematical Expression Recognition*

[4] Genthial G. & Sauvestre R., *Image to Latex*

[5] `https://github.com/harvardnlp/im2markup`

[6] `http://tc11.cvc.uab.es/datasets/CROHME-2014_2`

[7] `http://lstm.seas.harvard.edu/latex/data/`