

L^AT_EX-template for my dear Coli Friends

Luuk

25 oktober 2019

Inhoudsopgave

1	Introduction	1
2	Basics	1
2.1	A subsection	1
2.1.1	A subsubsection	1
2.2	<i>Style</i>	2
3	Math & Tables	2
4	Linguistics	2
4.1	IPA	2
4.2	Glossen	2
4.3	Syntax	3
5	Code	3
6	Bibliography	4

1 Introduction

Almost everyone in math/CS uses L^AT_EX to write up their documents. It looks really professional and you can make complex diagrams and formulae look really nice. The PDF for Probability Theory for example is very clearly made with L^AT_EX. Getting into this takes a bit of work, but you can play around with this template file, install other packages you may need etc.

2 Basics

2.1 A subsection

These are subsections

2.1.1 A subsubsection

Even smaller

2.2 *Style*

Lets include an image.



3 Math & Tables

Here is Lambda calculus and Fourier transform respectively. You can also see how examples work with gb4e.

(1)

$$(\lambda x.y)[y := x] = \lambda x.(y[y := x]) = \lambda x.x$$

(2)

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(x) e^{-2\pi i(n/T)x} dx$$

Tables are slightly cumbersome, but can look really nice. I recommend looking up the documentation, if you start using \LaTeX you will have to google a lot of things in the beginning anyways. Here's a table I made for a XOR neural network some time ago.

Input	Target	Output 1	Error 1	Output 3	Error 3
(1,1)	0	0,621	-0.621	0,0436	-0,0436
(1,0)	1	0,617	0,383	0,969	0,0308
(0,1)	1	0,617	0,383	0,972	0,0282
(0,0)	0	0,037	-0,037	0,0158	-0.0158

Tabel 1: XOR netwerk met 1 verborgen laag versus 3 verborgen lagen met 100000 epochs

4 Linguistics

4.1 IPA

Try to write your name in IPA with tipa.

(3) lyk sy:ɪmæɪəɪ

4.2 Glossen

This is a classic glossed linguistic example.

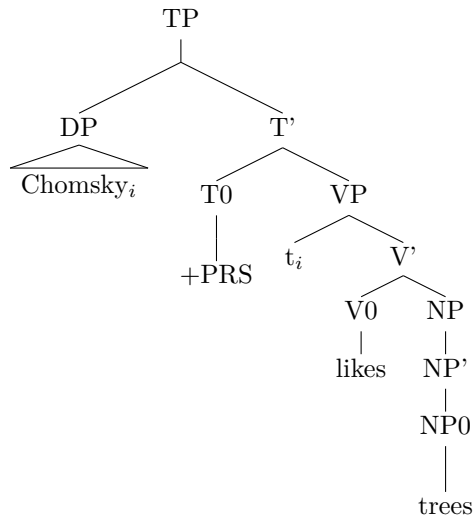
(4) Deze gloss-en zijn mooi
DEM.PROX gloss-PL BE.PL pretty
'These glosses are pretty'

4.3 Syntax

Let's draw a tree of (5) according to our favourite syntactic theory with QTREE.

(5) Chomsky likes trees

(6)



5 Code

Sometimes you want to show some code to your reader. Lets use the algorithm2a package to write some pseudocode. Let's then input some actual code from a file with inputlistings.

```
Result: Write here the result
initialization;
while While condition do
    instructions;
    if condition then
        instructions1;
        instructions2;
    else
        instructions3;
    end
end
```

Algorithm 1: How to write algorithms

```
1
2 #include <iostream>
3 #include <cmath>
4 #include <fstream>
5 #include <cstdlib>
6 #include <iomanip>
7 using namespace std;
8
9 const int MAX = 20;
10 const double ALPHA = 0.1;
11 const double BETA = 1.0;
12
```

```

13 // g-functie (sigmoid)
14 double g (double x) {
15     return 1 / ( 1 + exp ( - BETA * x ) );
16 }//g
17
18 // afgeleide van g
19 double gprime (double x) {
20     return BETA * g (x) * ( 1 - g (x) );
21 }//gprime
22
23 int main (int argc, char* argv[ ]) {
24
25     int inputs, hiddens;           // aantal invoer- en verborgen knopen
26     double input[MAX];           // de invoer is input[1]...input[inputs]
27     double inputtohidden[MAX][MAX]; // gewichten van invoerknopen 0..inputs
28                                     // naar verborgen knopen 1..hiddens
29     double hiddentooutput[MAX]; // gewichten van verborgen knopen 0..hiddens
30                                     // naar de ene uitvoerknoop
31     double inhidden[MAX];        // invoer voor de verborgen knopen 1..hiddens
32     double acthidden[MAX];       // en de uitvoer daarvan
33     double inoutput;             // invoer voor de ene uitvoerknoop
34     double netoutput;           // en de uitvoer daarvan: de net-uitvoer
35     double target;              // gewenste uitvoer
36     double error;               // verschil tussen gewenste en
37                                     // geproduceerde uitvoer
38     double delta;               // de delta voor de uitvoerknoop
39     double deltahidden[MAX];     // de delta's voor de verborgen
40                                     // knopen 1..hiddens
41     int epochs;                  // aant#include <iomanip>al trainingsvoorbeelden
42     int i, j, k;                 // tellertjes
43     //int seed = 1234;           // eventueel voor random-generator
44
45     if ( argc != 4 ) {
46         cout << "Gebruik: " << argv[0] << " <inputs> <hiddens> <epochs>" << endl;
47         return 1;
48     }//if
49     inputs = atoi (argv[1]);
50     hiddens = atoi (argv[2]);
51     epochs = atoi (argv[3]);
52     input[0] = -1;               // invoer bias-knoop: altijd -1
53     acthidden[0] = -1;          // verborgen bias-knoop: altijd -1
54     srand (time(NULL));
55
56 //etc...

```

6 Bibliography

There's 10 million ways to do bibliography in L^AT_EX depending on your citation style etc. etc. I will show you the easiest way, research your own way of doing it. This is not the most elegant way to do it, as the formatting is not automatic. I personally use the package biblatex. Anyways here are some references. I used [1], [2], [3] and [4]. The heading is in Dutch because of my language settings in this document.

Referenties

- [1] Gazzaniga, M. S., Ivry, R. B., & Mangun, G. R. (2002). *Cognitive neuroscience: The biology of the mind*. New York: Norton.
- [2] Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation* New York: Prentice Hall.
- [3] Russel, S.J., & Norvig, P. (2010). Learning in multi-layer networks. In *Artificial Intelligence: A modern approach*.(pp. 733-736) New Jersey: Prentice Hall.
- [4] Hansen, L.K., & Salamon, P. (1990). Neural network ensembles. *Transactions on Pattern Analysis and Machine Intelligence*, 12(80), 993-1001.