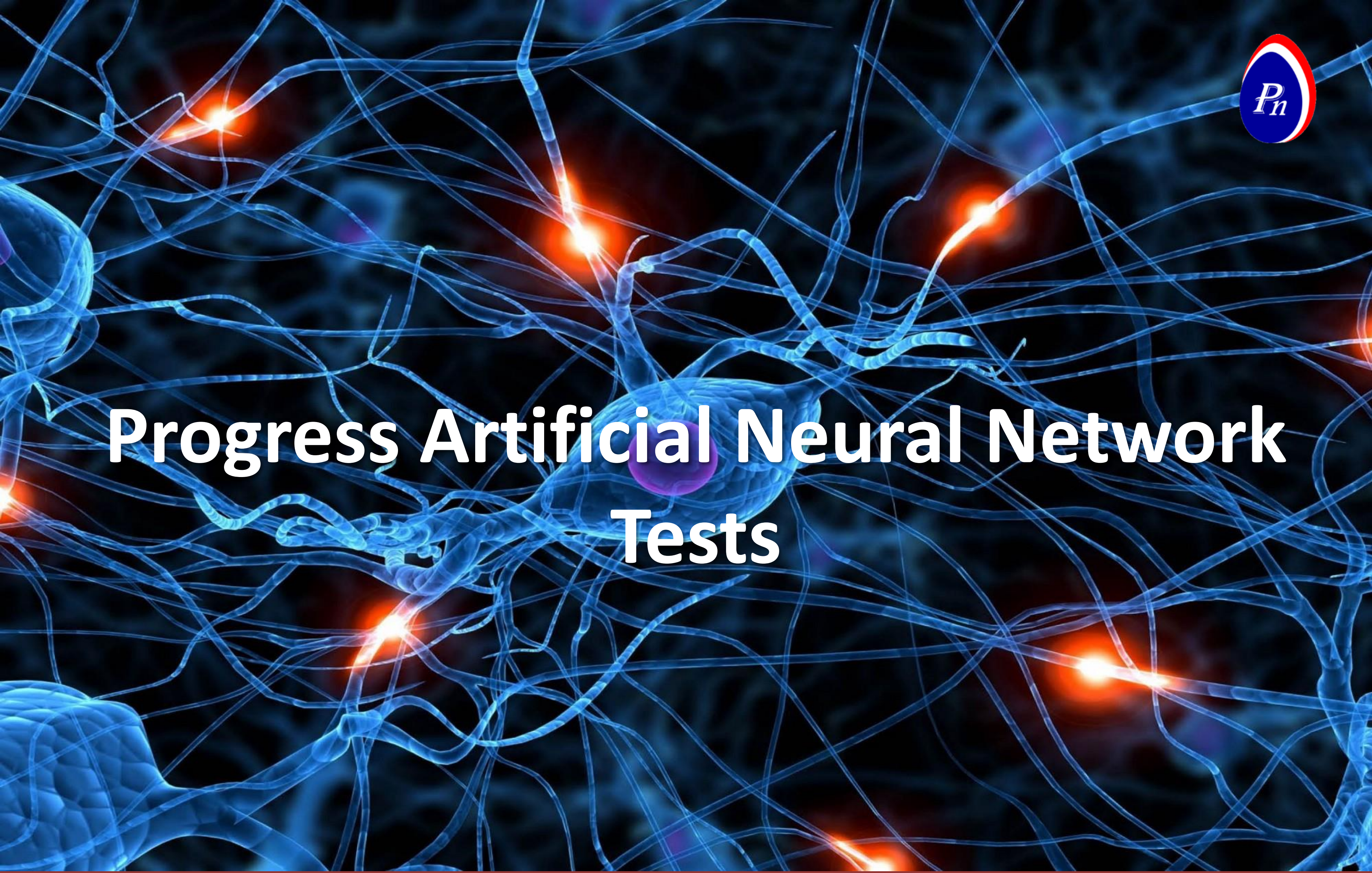


Progress Artificial Neural Network Tests

Progress, Inc.
Michigan, US
2020

Why experts are disappointed with existing ANNs

Francois Chollet, a leading practitioner of deep learning networks: ’**You cannot achieve general intelligence simply by scaling up today’s deep learning techniques.’**

- **Francois Chollet** , Google Artificial Intelligence Researcher, “Author of Keras - keras.io - Keras is a leading deep learning framework for Python, with over 200,000 users” See more at https://www.linkedin.com/in/fchollet/

Geoff Hinton, one of the most prominent AI scientists: ‘we need to start over’, when he explained that he was **’deeply suspicious’** of current AI techniques. **’My view is throw it all away and start again’.**

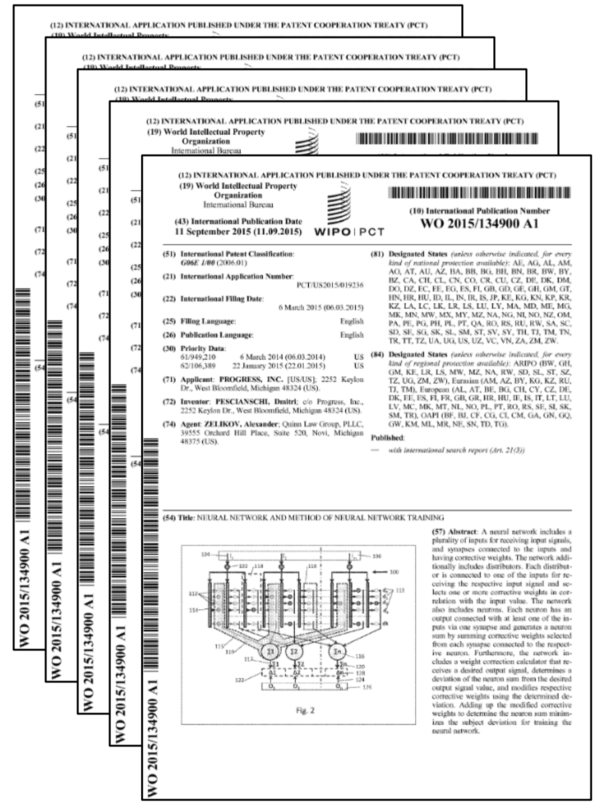
Geoffrey Everest Hinton is internationally distinguished for his work in artificial neural nets : Backpropagation, Boltzmann machine, Deep learning… He divides his time between his university (University of Toronto) research and his work at Google (Google Brain). See more at ttps://en.wikipedia.org/wiki/Geoffrey_Hinton

We indeed started it all from scratch - Progress, Inc. developed and patented a new neuron that resolved the core problems of Artificial Neural Networks.

Progress Intellectual Property

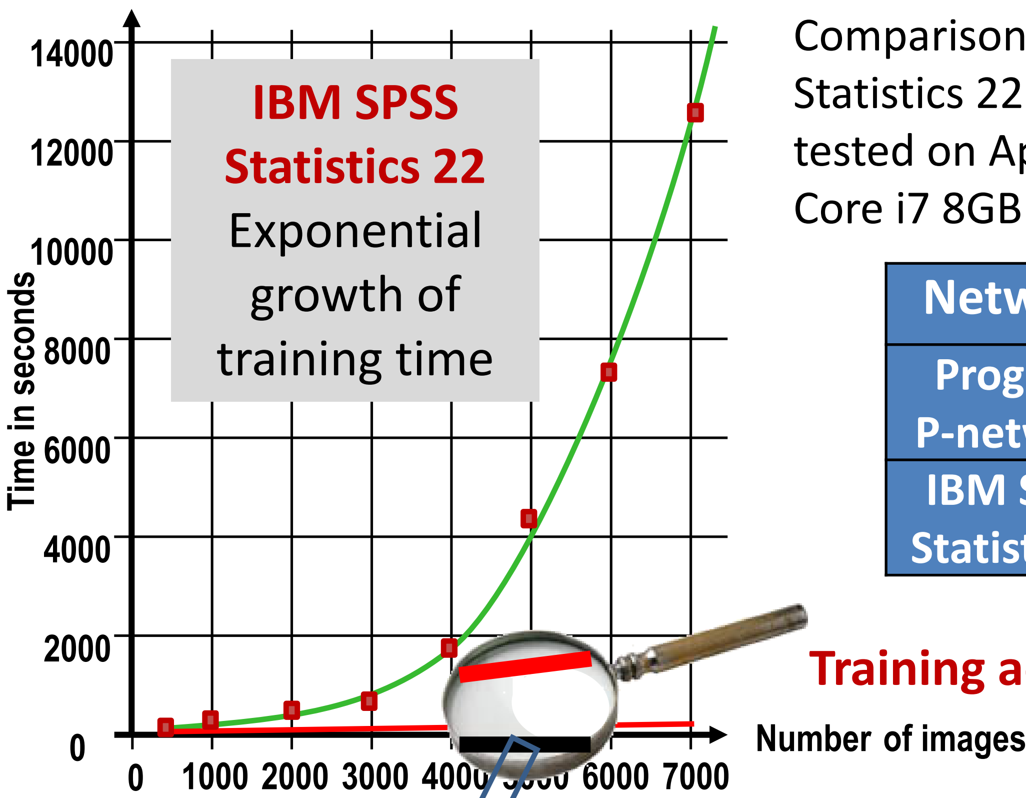
Progress current IP Portfolio includes the following patents:

1. US Patent No. 9390373; 2016
2. US Patent No. 9619749; 2017
3. US Patent No. 10423694; 2019
4. Japan Patent No. 6382354;2018
5. China Patent No. ZL201580012022.2; 2018
6. Mexico Patent No. MX357374B; 2018
7. Taiwan Patent No. I655587; 2019
8. Israel patent No. 247533; 2019
9. Hong Kong Patent No. HK1227499; 2019
10. Singapore Patent No. 11201608265X; 2019
11. Eurasian Patent No. 035114, 2020 (8 countries)
12. Patent allowance for Korean Patent Application No. 10-2016-7027849; 2020.



7 Patent Applications in 50 major countries (via Patent Cooperation Treaties) are filed and pending.

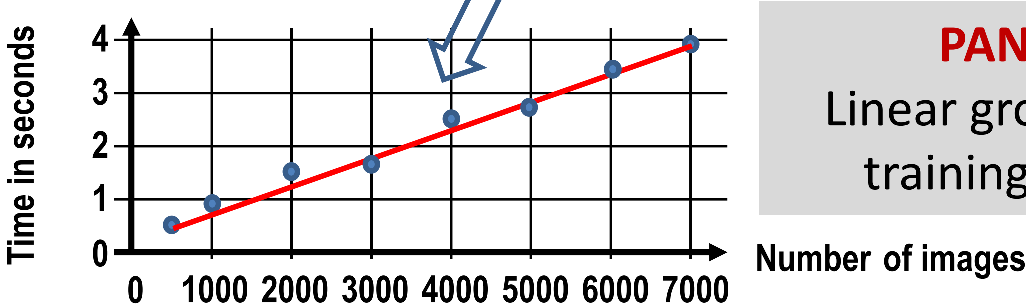
Example *of PANN testing*



Comparison of training time between IBM SPSS Statistics 22 and P-network, for the same problem, tested on Apple iMac 27" 3.5 GHz quad-core Intel Core i7 8GB of 1600MHz DDR3 memory; SSD

Network	Images	Training time
Progress P-network	7000	4 sec.
IBM SPSS Statistic 22	7000	13400 = 3h. 43min.

Training advantage factor $= \frac{13400\ sec}{4\ sec} = 3350$



PANN
Linear growth of training time

Both nets are tested on the same statistic data.

Training Time and Error Benchmarking

Target error: 0.02

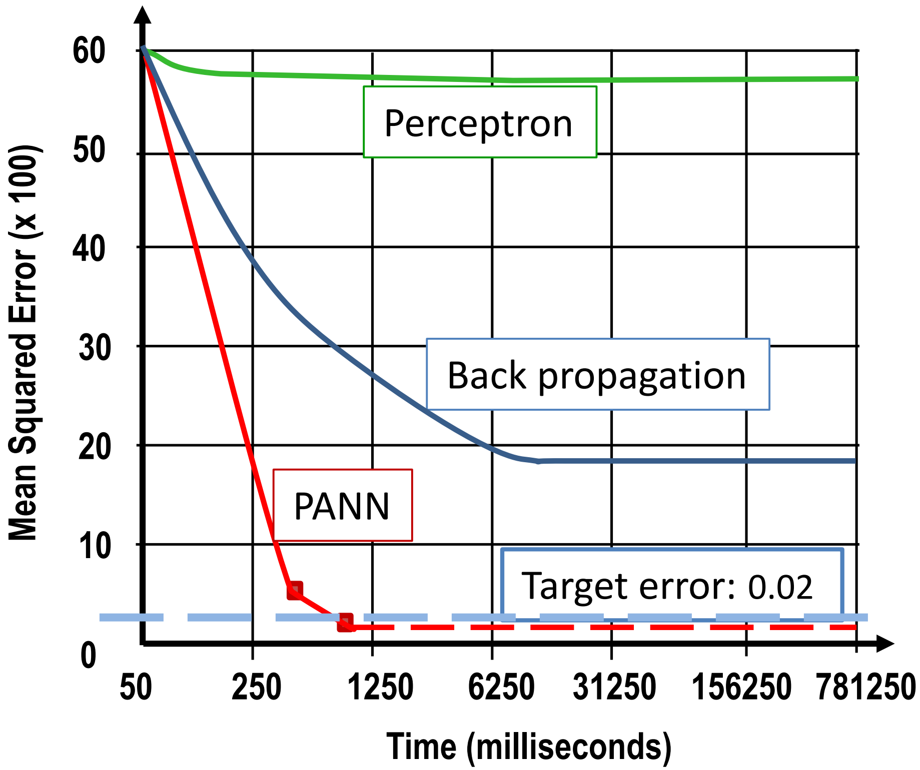
1					
2			Epoch	Error	Training Time (ms)
3	8/24/2017 10:34:41 AM: Train on: C:\User	Perceptron network frame1:	1	0.570838	156
4	8/24/2017 10:35:11 AM: Train on: C:\User	Perceptron network frame1:	5	0.570126	422
5	8/24/2017 10:35:18 AM: Train on: C:\User	Perceptron network frame1:	10	0.568272	733
6	8/[illegible]:\User	Perceptron network frame1:	15	0.56775	1046
7	8[illegible]User	Perceptron network frame1:	20	0.567693	1451
8	8/24/2[illegible]n: C:\User	Perceptron network frame1:	30	0.567577	2091
9	8/24/2017 10:35:52 AM: Train on: C:\User	Perceptron network frame1:	40	0.567287	2746
10	8/24/2017 10:36:05 AM: Train on: C:\User	Perceptron network frame1:	50	0.567535	3261
11	8/24/2017 10:36:37 AM: Train on: C:\User	Perceptron network frame1:	100	0.567109	6350
12					
13	8/24/2017 10:37:23 AM: Train on: C:\User	Back propagation network frame1:	1	0.194059	7333
14	8/24/2017 10:38:44 AM: Train on: C:\User	Back propagation network frame1:	5	0.190227	32573
15	8/24[illegible]\User	Back propagation network frame1:	10	0.189882	72556
16	8[illegible]er	Back propagation network frame1:	15	0.189822	108296
17	8[illegible]er	Back propagation network frame1:	20	0.189851	139792
18	8/24[illegible]:\User	Back propagation network frame1:	30	0.189813	211678
19	8/24/2017 10:5[illegible]rain on: C:\User	Back propagation network frame1:	40	0.18993	287713
20	8/24/2017 11:02:52 AM: Train on: C:\User	Back propagation network frame1:	50	0.189789	367179
21	8/24/2017 11:16:43 AM: Train on: C:\User	Back propagation network frame1:	100	0.189803	734109
22					
23	8/24/2017 11:21:21 AM: Train on: C:\User	Progress P-network frame1:	1	0.058916	343
24	8/24/2017 11:22:01 AM: Train on: C:\User	Progress P-network frame1:	5	0.018097	640
25	8/24/201[illegible]n: C:\User	Progress P-network frame1:	10	0.018097	640
26	8/[illegible]User	Progress P-network frame1:	15	0.018097	671
27	8[illegible]ser	Progress P-network frame1:	20	0.018097	3640
28	8/2[illegible]User	Progress P-network frame1:	30	0.018097	9252
29	8/24/2017 11:2[illegible]n on: C:\User	Progress P-network frame1:	40	0.018097	12262
30	8/24/2017 11:24:15 AM: Train on: C:\User	Progress P-network frame1:	50	0.018097	14702
31	8/24/2017 11:25:00 AM: Train on: C:\User	Progress P-network frame1:	50	0.018097	15631
32	8/24/2017 11:25:55 AM: Train on: C:\User	Progress P-network frame1:	100	0.018097	30670

Perceptron training

Back propagation training

PANN training

Benchmarking of Training Time and Training Error



Progress ANN reaches desired minimum error in less than one second.

Perceptron and Back Propagation ANNs have very large error, which reduces slowly.

There is no tendency for Perceptron and Back Propagation ANN to reach target error.

Testing training set:

- 30,000 images

Benchmarking of DNN and PANN based on Iris Flower data set

Progress Artificial Neural Network (PANN) was compared with **Deep Learning ANN (DNN)** based on advanced **Google TensorFlow** technology and with help of the standard statistical **Iris Flower data set** (120 training samples and 30 testing samples).

Iris Data Set
https://en.wikipedia.org/wiki/Iris_flower_data_set
http://archive.ics.uci.edu/ml/datasets/Iris

Benchmarking results:

1. PANN training speed is 3,000 – 5,000 times higher than DNN training speed
2. PANN training error is less or equal to DNN best testing results.

Benchmarking of DNN and PANN based on Iris Flower data set (Test Screenshot)

```
C:\Python36>python.exe iris_tf.py

Load datasets from iris_training.csv 120 samples and iris_test.csv 30 samples
Build DNN with 3 hidden layers |10|20|10|
Training DNN Steps = 50
2018-01-24 15:19:07.862516: I C:\tf_jenkins\home\workspace\rel-win\M\windows\PY\
36\tensorflow\core\platform\cpu_feature_guard.cc:137] Your CPU supports instruct
ions that this TensorFlow binary was not compiled to use: AVX
Testing DNN on 30 patterns from iris_test.csv

Train time 11306 ms. Test accuracy: 0.933333


C:\Python36>python.exe iris_ml.py

Load train dataset iris_training.csv
Create MLP Classifier
Load test dataset iris_test.csv
Testing 21 [ 5.6  3.   4.5  1.5] , 2.0 != 1.0
Testing 23 [ 6.3  2.5  4.9  1.5] , 2.0 != 1.0
Recognized 28.0 in 30 patterns from test dataset iris_test.csv

Train time 215 ms.  Test accuracy: 0.933333


C:\Python36>python.exe iris_pn.py

Create P-Net && load train dataset iris_training.csv
Information P-Net Layers=2 |4|1| Input=4 Output=1 Intervals=10
Training P-Net Patterns=120 Epoch=3 Error=0.078208 Time=0 ms.
Testing 7 [6.2999999999999998, 2.7999999999999998, 5.0999999999999996, 1.5] , 1
!= 2
Recognized 29.0 in 30.0 patterns from test dataset iris_test.csv
Save P-Net configuration to .\iris_pn\iris.nnw

Train time 2 ms.  Test accuracy: 0.966667
```

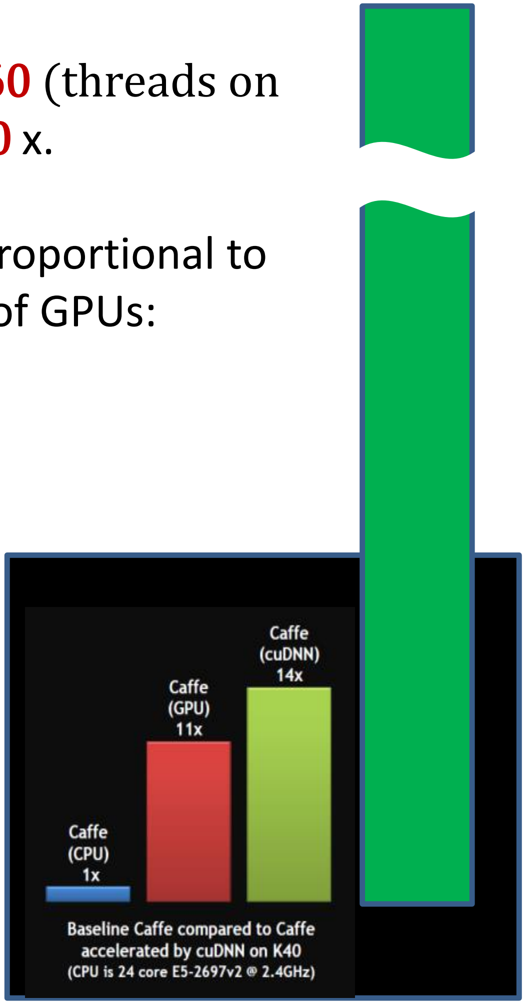
GPUs breakthrough

Comparison of PANN training speed and nVidia cuDNN

PANN demonstrates a thousand times higher training speed on CPU and GPUs than existing ANN.

PANN provides 60 (threads on GPU) x = 50 000 x.

Acceleration is proportional to the number (N) of GPUs:
50 000 x N



Permits to:

- Improve ANN training speed thousands of times
- Build supercomputer on GPUs
- Build Hypercomputer on GPUs