Math INDUSTRY

Analyzing deforestation in the Amazon Basin with Machine Learning

MULTIVERSE

The Team

Mehdi Bozzo-Rey
CRO, Multiverse

Sukhi Singh
Principal Scientist, Multiverse

Santanil Jana
PhD, UBC

Youssef Mousaaid
Post doc, Perimeter Institute

Reza (Daniel) Sadoughian
MSc, UBC

Manish Krishan Lal
PhD, UBC

Kimathra Reddy
BSc, UBC

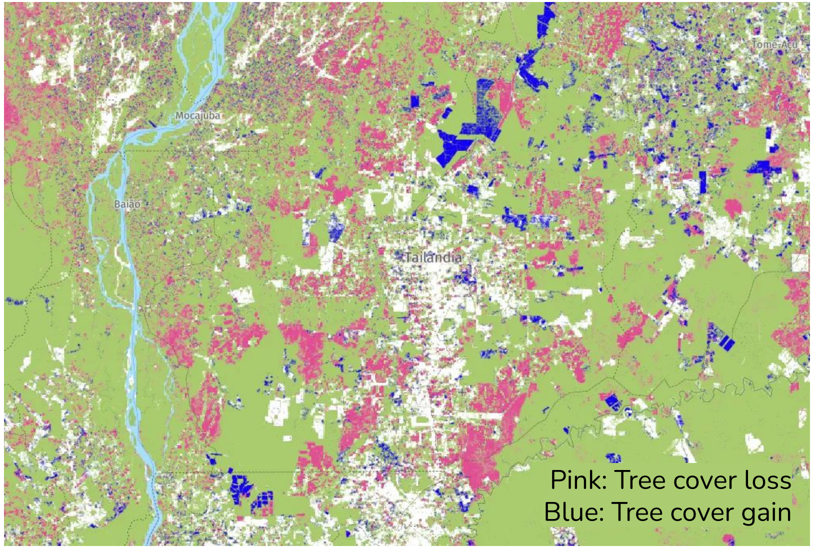
C Shijia Yu
MSc, UToronto

The Forest Degradation Problem

- ➢ Every minute, the world loses an area of forest the size of 48 football fields

How to reduce forest degradation?

- ➢ Early detection of forest destruction
- ➢ Identification of lands where trees can be replanted
- ➢ Detection and monitoring of patterns



Changes in tree cover in Tailândia municipality in the Amazon rainforest — Global Forest Watch.

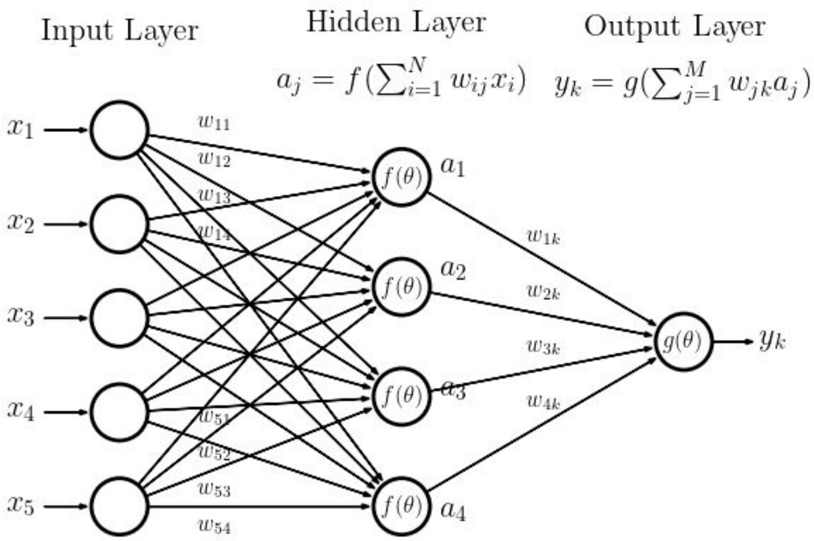
Kaggle's Multi-label Classification Competition *Planet: Understanding the Amazon from Space*

- **4 Atmospheric conditions:** clear, partly cloudy, cloudy, haze
- **7 Common land cover and land use types:** rainforest, agriculture, rivers, towns/cities, roads, cultivation, bare ground
- **6 Rare land cover and land use types:** slash and burn, selective logging, blooming, conventional mining, artisanal mining, blow down

Neural Networks 101

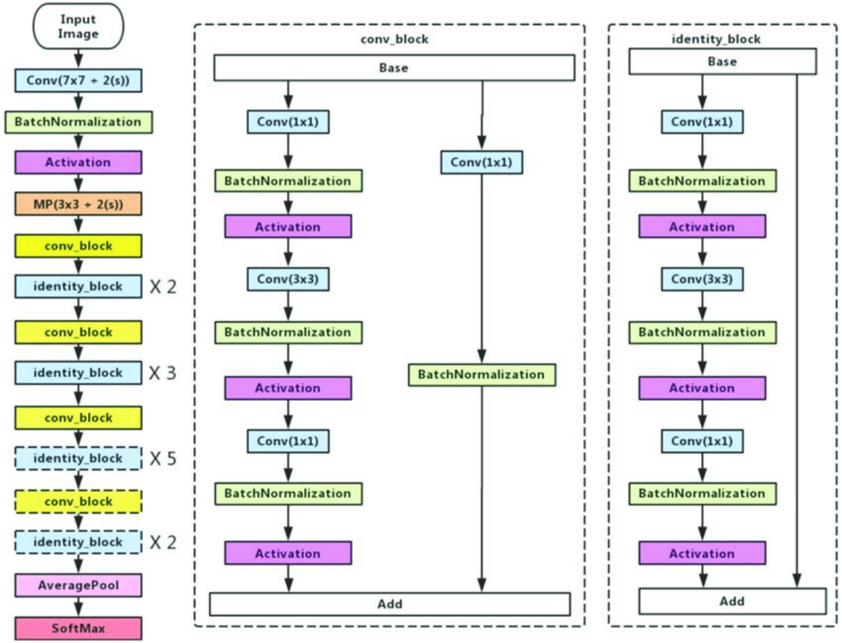
- **Input Layer:** Receives and passes data to hidden layers.
- **Hidden Layer(s):** Performs computations and feature extractions.
    - Uses matrix-vector multiplication
- **Output Layer:** Produces the final prediction or output.
    - In our case, there will be 17

Training a neural network means minimizing the loss (cost) function and involves calculus techniques like gradient descent.



A hidden layer is any layer between the input layer and output layer. There can be multiple hidden layers!

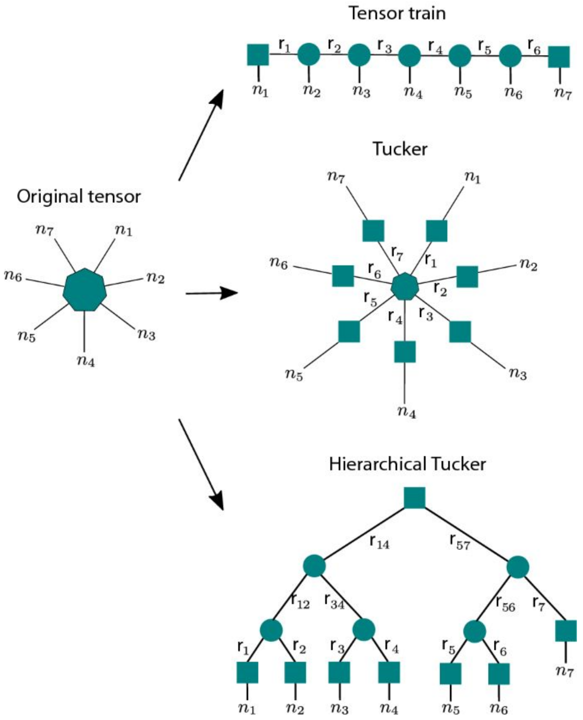
ResNet



The ResNet50 Architecture

- ## ResNet-50:
  - A deep CNN architecture, 50 layers.
  - Combining depth and scale for performance.
- ## The Significance of Depth:
  - Deeper Networks enable capturing complex features.
  - Avoids the vanishing gradient problem.
- ## Unveiling the Scale:
  - Multiple scales of features through skip connections.
  - Empowering hierarchical representation learning.
- ## The Challenge of Size:
  - Over 25 million parameters.
  - Long training times and large computations.
- ## Environmental Impact:
  - Massive computational resources needed.
  - Contributes to increased carbon footprint.

Tensor Decomposition



Tensor Decompositions

➢ Tensor Decomposition:

- A powerful technique to represent and analyze high-dimensional tensors.
- Breaking multi-dimensional arrays into simpler components (rank).

➢ TT Decomposition:

- Representing a tensor as a sequence of smaller, interconnected tensors.
- Compressing large tensors into a compact format.
- Memory efficiency: Reducing storage requirements for big data tensors.

➢ Tensor Train in Deep Learning:

- Accelerating computation in neural networks.
- Efficiently handling large model parameters.

Compression of a CNN model

- By replacing the output layer of ResNet50 with *17* neurons, we obtain *32,768*17* new weight matrix parameters and *17* bias vector parameters, totaling *557,073* new parameters.
- Just to retrain the final layer of our model (i.e. training the *557,073* parameters) for *20* epochs with GPU on Google Colab it takes around *40* minutes.
- Using TT decomposition on the weight matrix on the output layer of the ResNet50 architecture we are able to reduce the number of parameters to only *17,425*.
- Retraining the new model with the same setup takes less time and gives comparable accuracy.
- Tensorizing the entire model using TT decomposition gives significant advantages in terms of speeding up training time and also utilization of resources.
- Complete tensorization using TT decomposition will further reduce training time and model size, enabling implementation on previously inaccessible devices.

Setup

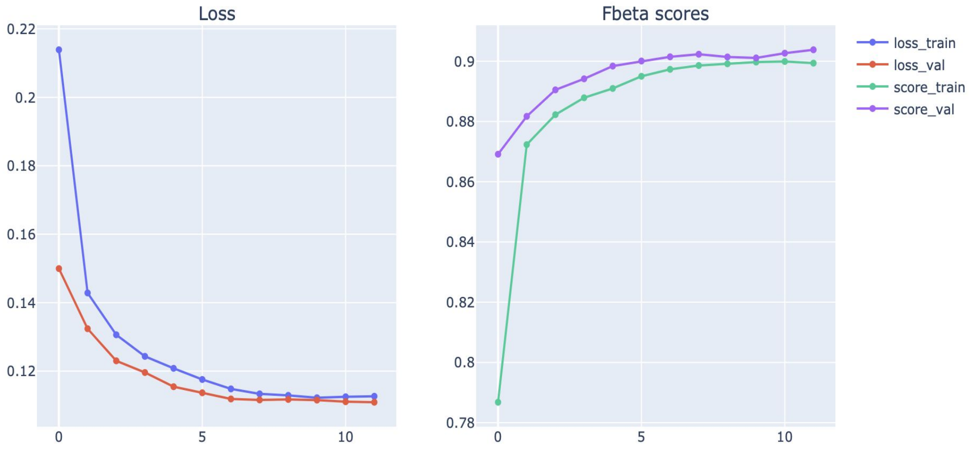
➢ **Baseline:**

ResNet50 + FC1(2048,512) + FC2(512,17)

➢ **Number of parameters of FC1:** 1048576

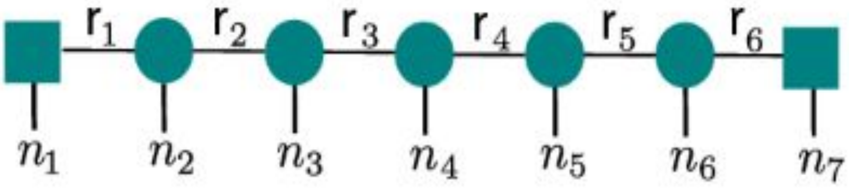
➢ **Number of epochs:** 12

FC1 layer --- number of parameters: 1048576

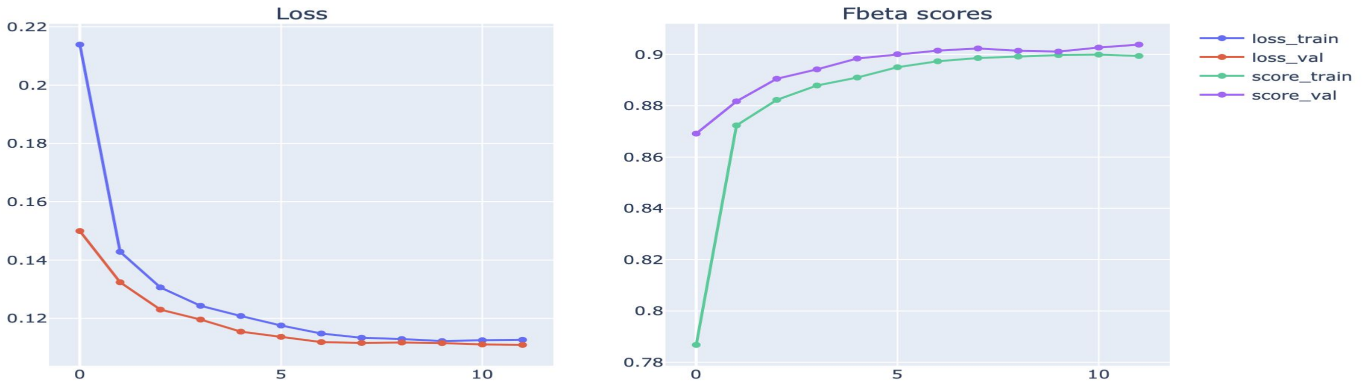


Tensorizing the fully connected layer FC1(2048,512)

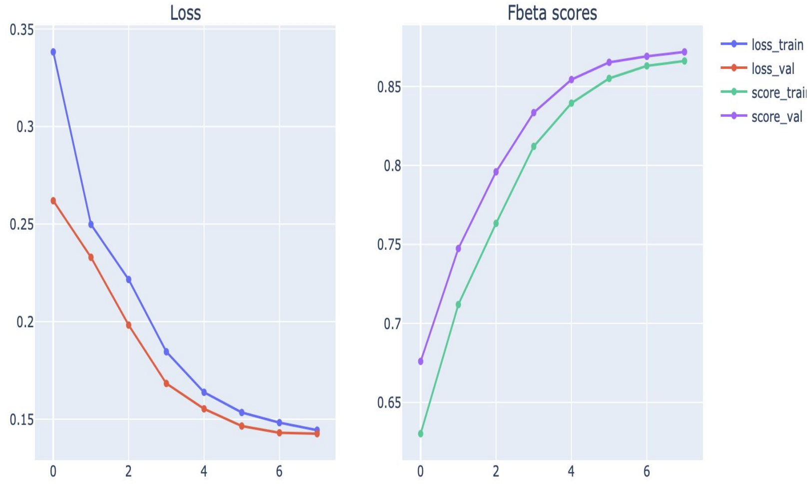
- Reshape the weight matrix 2048x512
- TT decomposition of the weight matrix
- Train the network with various tt-ranks=(r1,...,r6).



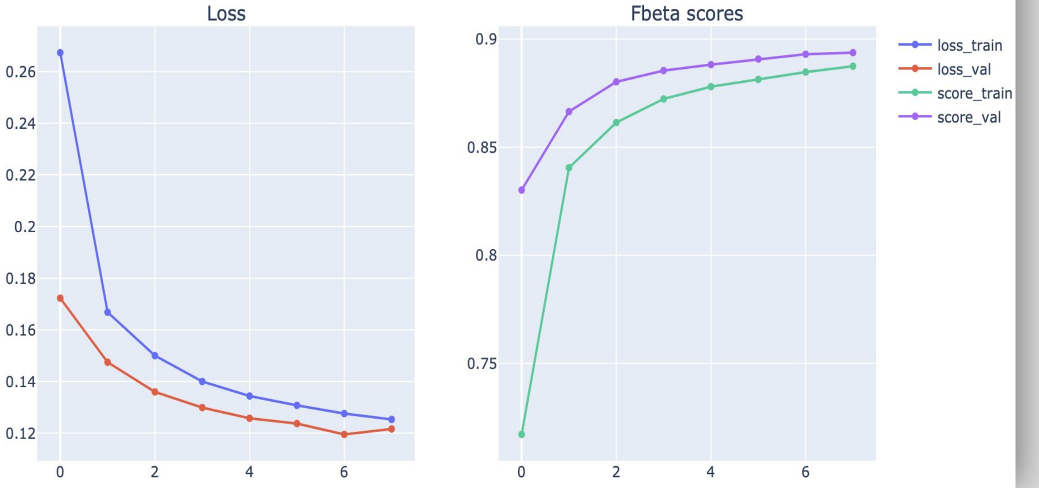
FC1 layer --- number of parameters: 1048576



Shape: (16,8,16,512) --- tt-rank=(24,24,24,24) --- number of parameters: 345024



Shape: (32,64,512) --- tt-rank=(64,64,64) --- number of parameters: 301056



Experimental results

Model	Comp. ratio	Accuracy	Inference time/epoch(s)
Baseline ResNet18+FC	1:1	89.00%	847
(8x8x8), tt-rank=(8,8,8,8)	1:12	71.00%	859
(8x8x8), tt-rank=(16,16,16,16)	1:1	83.12%	851
Baseline ResNet50+FC	1:1	**90.30%**	929
(16x8x16), tt-rank=(24,24,24,24)	**1:3**	87.20%	935
(16x8x16), tt-rank=(32,32,32,32)	1:1	**89.09%**	939
(32x64), tt-rank=(64,64,64)	**1:3.5**	**89.37%**	929

Why tensorize

- State-of-the-art deep neural networks are too big to fit into the memory of mobile devices.
- There is lot of redundancy in big neural networks
- Tensorization allows the reduction of the size of FC with a small drop in accuracy.
- The ultimate goal is to increase accuracy while decreasing memory and complexity.

Thank you!