At my company we are starting an experimental project to extend EVM with basic deep learning capabilities.

This is not to train a neural network, but to use a pre-trained neural network inside a smart contract. Computation-wise using a pre-trained neural network is actually not so much more expensive than doing, say, RSA.

I understand that this may be a bit too heavy for the official Ethereum blockchain, so in our case we will run the EVM on a separate permissioned cluster with BFT-like consensus.

The current plan is that:

1. A pre-trained network is saved on the blockchain. We can use some of existing

neural network serialization standards such as the ones used inKeras framework

1. The EVM will need to pull the neural network from the blockchain.
2. In the simplest case there we add a single predict

instruction similar to predict from Keras framework. This instruction will take a fully qualified name of the neural network and an input data array, run the neural network and produce output data.

As an example input data could be an English-language string, and output will be a German translation of this string.

One problem that we will need to solve in the process is introducing deterministic floating point numbers such as IEEE 754-2008 into the EVM in some way.

If there are other people interested to run AI on EVM, we would be willing to cooperate on this to establish a standard that everyone uses …