On the asymptotic outcomes of generative AI

Boualem Djehiche (KTH, Stockholm)

November 2023

Context

- GPT: Generative pre-trained transformer
- BloombergGPT is specialized to **economic time series** (see Wu *et al.* (2023).)
- It is a 50 billion parameter language model that is trained on a wide range of **economic data**
- It contains 363 billion token dataset based on Bloomberg's extensive data sources, perhaps the largest domain-specific dataset yet, augmented with 345 billion tokens from general purpose datasets
- Key ingredients: Data+Neural Network

Neural Network

The neural network specification is given by the family

$$\mathbb{NN} := (x_0, R, W, b, L, \{\mathcal{H}_l\}_{1\leq l\leq L})$$

with

- $L \geq 2$ layers, $\{\mathcal{H}_l\}_{0\leq l\leq L}$ non-zero Hilbert spaces.
- $\forall\, t \in \mathbb{N}$, $\forall l \in \{1, \ldots, L\}$,
  - $W_{l,t} : \mathcal{H}_{l-1} \to \mathcal{H}_l$ a bounded linear operator and activation operator
  - $R_{l,t} : \mathcal{H}_l \to \mathcal{H}_l$.
- $W_{l,t}$ is the weight operator and $b_{l,t}$ captures the bias parameter.
- The shallow/deep neural network is the map

$$R_{L,t} \circ (W_{L,t} \cdot + b_{L,t}) \circ R_{L-1,t} \circ (W_{L-1,t} \cdot + b_{L-1,t}) \circ \ldots \circ R_{1,t}(W_{1,t} \cdot + b_{1,t}).$$

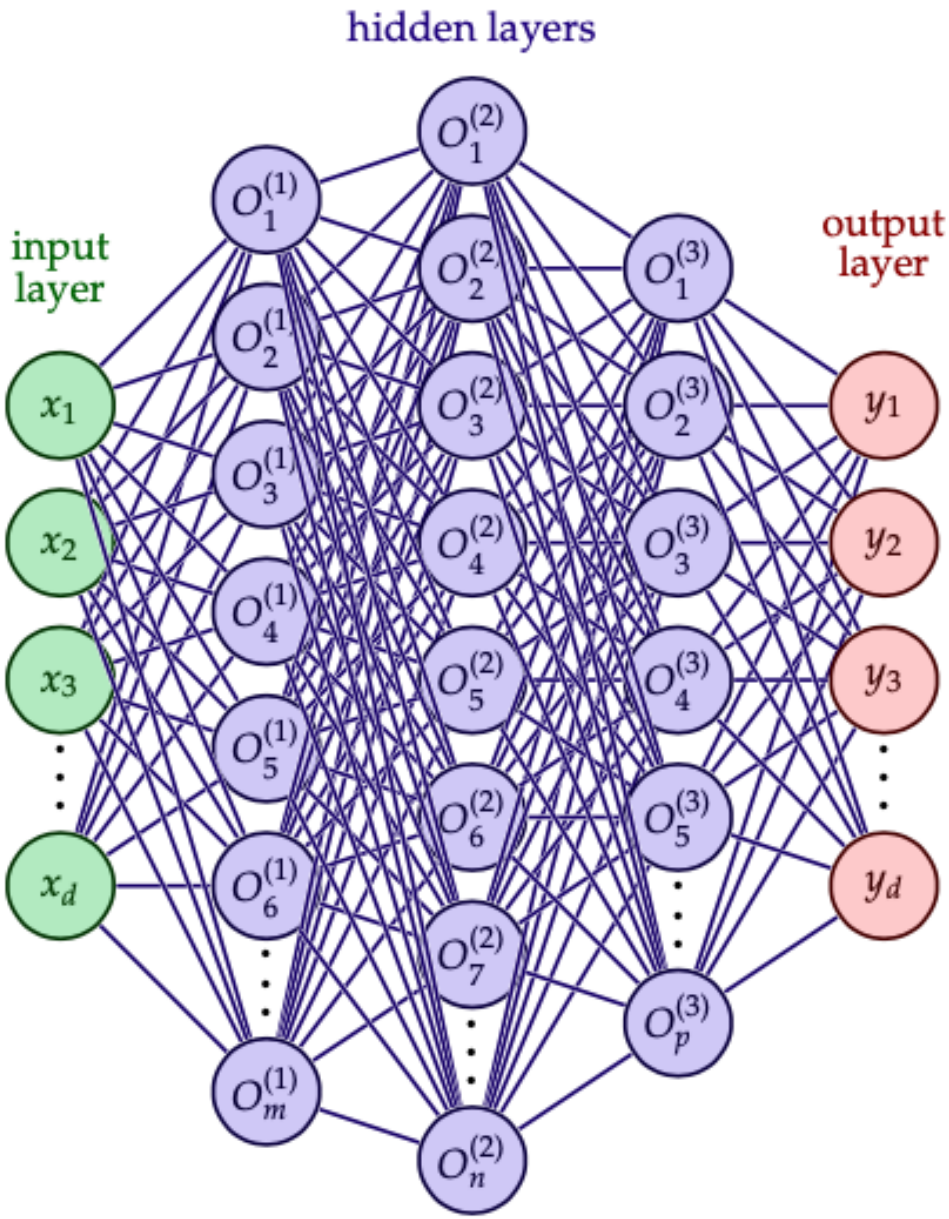
Input-Output

The input-output at layer l is given by the map defined by

$$\begin{aligned} O_{l,t} : \mathcal{H}_{l-1} &\to \mathcal{H}_l \\ x &\mapsto R_{l,t}(W_{l,t}x + b_{l,t}), \end{aligned} \tag{1}$$

Given the input x_t, the output of the (entire) neural network at time t is

$$y_{L,t} = O_{L,t} \circ O_{L-1,t} \circ \ldots \circ O_{1,t}(x_t) \quad \text{with} \quad y_{l,t} = O_{l,t}(y_{l-1,t}).$$


hidden layers
input layer
output layer
x_1
x_2
x_3
x_d
$O_1^{(1)}$
$O_2^{(1)}$
$O_3^{(1)}$
$O_4^{(1)}$
$O_5^{(1)}$
$O_6^{(1)}$
$O_m^{(1)}$
$O_1^{(2)}$
$O_2^{(2)}$
$O_3^{(2)}$
$O_4^{(2)}$
$O_5^{(2)}$
$O_6^{(2)}$
$O_7^{(2)}$
$O_n^{(2)}$
$O_1^{(3)}$
$O_2^{(3)}$
$O_3^{(3)}$
$O_4^{(3)}$
$O_5^{(3)}$
$O_p^{(3)}$
y_1
y_2
y_3
y_d

Question 1: What does a well-trained Deep Neural Network Do?

- Find and characterize the asymptotic behavior (as $t \to \infty$) of the network for a large class of architectures $\mathbb{NN}$ under the constraint $\mathcal{H}_0 = \mathcal{H}_L$.

- Characterize the possible limit (if any) of $\lim_{t \to \infty} (x_t, y_{1,t}, \ldots, y_{L,t})$.

Iterated input-output

Iterate layer by layer and by timestep the following:

$$\text{for } t \in \{0, 1, 2, \ldots\} \;\Big|\; \begin{array}{l} y_{1,t} = O_{1,t}(x_t), \\ \text{for } l \in \{2, \ldots, L\},\ y_{l,t} = O_{l,t}(y_{l-1,t}), \\ x_{t+1} = y_{L,t} \end{array} \tag{2}$$

Here the output (in $\mathcal{H}_L$) of the neural network is used as input (feedback) to the input layer for the next iteration. This is the commonly used input-output feed!

Asymptotic outcomes

For the commonly used activation functions $R_{\ell,t} := r_\ell$, most of the time

$$\lim_{t \to +\infty} (x_t, y_{1,t}, \ldots, y_{L,t}) \qquad \text{diverges!}$$

Iterated input to averaged output

We suggest the following alternative input-output feed.

Let $\lambda_t \geq 0$, Iterate layer by layer and by timestep the following:

$$\text{for } t \in \{0,1,2,\ldots\} \;\Big|\; \begin{array}{l} y_{1,t} = O_{1,t}(x_t), \\ \text{for } l \in \{2,\ldots,L\},\ y_{l,t} = O_{l,t}(y_{l-1,t}), \\ x_{t+1} = x_t + \lambda_t(y_{L,t} - x_t). \end{array} \tag{3}$$

Here, the **averaged output** of the neural network is used as input for the next iteration.

Q1: Asymptotic outcomes

$$\lim_{t\to+\infty} (x_t, y_{1,t}, \ldots, y_{L,t})?$$

Question 2: Training & Design

Given an input-output data set

$$\{(x_t, d_t = y_{L,t}),\ t \in \{1, \ldots, T\}\},$$

Find $\mathbb{NN}$ such that

$$O_{L,t} \circ O_{L-1,t} \circ \ldots \circ O_{1,t}(x_t) = d_t, \qquad \text{for each } t.$$

Training & Design: reformulated

Question 2 is reformulated in a weak sense as a variational inequality.

Training as a variational inequality

Find a vector $\theta^* = (W^*, b^*)$ such that

$$\sum_{t=1}^{T} \omega_{l,t} \Big\langle (R_{l,t}[A_{l,t}\theta^*_{l,t}] - y_{l,t}), A_{l,t}(\theta - \theta^*_{l,t}) \Big\rangle \geq 0, \quad \forall \theta, \ \forall l \in \mathcal{L}, \tag{4}$$

$\omega_{l,t} \geq 0$, $\sum_{t=1}^{T} \omega_{l,t} = 1$, $A_{l,t} : (W_{l,t}, b_{l,t}) \mapsto W_{l,t}x_l + b_{l,t} := A_{l,t}\theta_{l,t}$ is linear and non-zero.

Moreover, design an algorithm that approximates θ^*, the solution (if any) to (4).

Activation functions used in Deep Learning

We note the following about the 40 activation functions $R_{k,t} := r_k$ used in Deep Learning.

- Their compositions are **not** necessarily convex. (eg. $ReLU_1 \circ ReLU_2$ is not convex)
- They are **not** necessarily strict contraction maps.
- **Interestingly, 40 of the most used activation functions implemented in the deep learning literature are γ-averaged** for some $0 < \gamma \leq 1$.

Definition (γ-averaged operator)

• Let $\gamma \in (0, 1]$. An operator $O : \mathcal{H} \to \mathcal{H}$ is γ-averaged if $[Id + \frac{1}{\gamma}(O - Id)]$ is 1-Lipschitz continuous.

• The composition $O_L \circ O_{L-1} \circ \dots O_1 : \mathcal{H} \to \mathcal{H}$ of γ_l-averaged operators O_l is $\frac{1}{1 + \frac{1}{\sum_{l=1}^{L} \frac{\gamma_l}{1-\gamma_l}}}$-averaged.

Weak convergence to the set of fixed-points

To address these questions, the first tool we will use is the following

Theorem (Bauschke *et al.* (2011), Baillon *et al.* (2012))

- *Let* $0 < \gamma \leq 1$, *and* $O : \mathcal{H}_0 \to \mathcal{H}_0$ *be a* γ*-averaged operator such that*

$$\mathit{Fixed}(O) = \{x \in \mathcal{H}_0 |\, x = O(x)\} \neq \emptyset.$$

- *Let* $\{\lambda_t\}_{t\geq 1}$ *be a sequence in* $(0, \frac{1}{\gamma})$ *such that* $\sum_{t\geq 1} \lambda_t(1 - \gamma \cdot \lambda_t) = +\infty$.
- *Assume that* $x_0 \in \mathcal{H}_0$ *and set*

$$x_{t+1} = x_t + \lambda_t(O(x_t) - x_t). \qquad \textit{(cf. the Krasnoselskii-Mann scheme)}$$

Then $O(x_t) - x_t$ *converges to* 0 *as* $t \to \infty$. *Moreover,* x_t *converges weakly to a point in* $\mathit{Fixed}(O)$.

This theorem is related to the so-called method of alternating projections that goes back to Schwarz' alternating method in PDEs. Also, it (partly) answers the question whether limit cycles can be characterized as the minimizers of a certain functional.

Exploiting the hidden convexity part of activation functions

The second tool is to exploit the convexity part of the activation functions.

By Moreau's Theorem, for $\gamma < 1$, any γ-averaged map r_k can be rewritten as

$$r_k = [Id + \partial f_k]^{-1} = [\partial(\frac{1}{2}\| \cdot \|^2 + f_k)]^{-1}.$$

The Legendre-Fenchel duality yields

$$[\partial(\frac{1}{2}\| \cdot \|^2 + f_k)]^{-1} = \partial[(\frac{1}{2}\| \cdot \|^2 + f_k)^*],$$

where $\psi^*(x) = \sup_y[\langle x, y\rangle - \psi(y)]$ is the Legendre-Fenchel conjugate of ψ.

The activation functions that are γ-averaged ($\gamma < 1$) are subgradients

Therefore, for each $k \in \{1, \dots, 40\}$, we have

$$r_k = \partial[(\frac{1}{2}\| \cdot \|^2 + f_k)^*] := \partial g_k. \tag{5}$$

Table: Activation functions that are γ-averaged, $\gamma \in (0, 1]$.

Activation function	Expression	γ-averagedness
Identity	$r_1(x) = Id(x) = x$	1
Linear	$r_2(x) = \langle \lambda, x \rangle + b$	$\frac{(1+\|\lambda\|_\infty)}{2}\mathbb{I}_{\{\|\lambda\|_\infty \leq 1\}}$
Rectified linear unit (ReLU)	$r_3(x) = \max(0, \langle \lambda, x \rangle + b)$	$\frac{(1+\|\lambda\|)}{2}\mathbb{I}_{\{\|\lambda\| \leq 1\}}$
Logistic (Sigmoid, Soft step)	$r_4(x) = \frac{1}{1+exp(-\langle \lambda, x \rangle - b)}$	$\frac{(4+\|\lambda\|)}{8}\mathbb{I}_{\{\|\lambda\| \leq 1\}}$
Sigmoid	$\sigma(x) = sigmoid(x) = \frac{1}{1+e^{-x}}$	$\frac{5}{8}$
Hyperbolic tangent	$r_5(x) = \lambda \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$\frac{(1+\|\lambda\|)}{2}\mathbb{I}_{\{\|\lambda\| < 1\}}$
Softmax	$r_6(x) = \frac{e^{\lambda x_i}}{\sum_{k=1}^K e^{\lambda x_k}}$	$\frac{(1+\|\lambda\|)}{2}\mathbb{I}_{\{\|\lambda\| < 1\}}$
Gaussian error linear unit (GELU2)	$r_7(x) = \lambda \frac{1}{2} x \left[1 + tanh\left(\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)\right)\right]$	$\frac{18}{20}$

Table: Activation functions that are γ-averaged, $\gamma \in (0, 1]$ (cont.)

Gaussian error linear unit	$GELU(x) = \lambda(x)\mathbb{P}(X \leq x) = \lambda(x)\frac{1}{2}\left[1 + erf(\frac{x}{\sqrt{2}})\right], \quad X \sim \mathcal{N}(0,1)$	$\frac{(1+\|\lambda\|)}{2}\mathbb{I}_{\{\|\lambda\|\leq 1\}}$
Softplus	$r_8(x) = \lambda \log(1 + e^x)$	$\frac{(1+\|\lambda\|)}{2}\mathbb{I}_{\{\|\lambda\|\leq 1\}}$
Softplus	$softplus(x) = \frac{1}{\lambda}\log(1 + e^{\lambda x})$	1
	$softplus(x) = \log(1 + \sum_{k=1}^{d} e^{x_k})$	1
Exponential linear unit (ELU)	$r_9(x) = \lambda(e^x - 1)\mathbb{I}_{\{x\leq 0\}} + x\mathbb{I}_{\{x>0\}}$	1
Scaled exponential linear unit	$r_{10}(x) = \lambda[\alpha(e^x - 1)\mathbb{I}_{\{x<0\}} + \alpha x\mathbb{I}_{\{x\geq 0\}}], \quad (\alpha, \lambda) = (0.0507, 0.6733)$	$\lambda\alpha$
Leaky rectified linear unit	$r_{11}(x) = 0.01x\mathbb{I}_{\{x<0\}} + x\mathbb{I}_{\{x\geq 0\}}$	1

Table: Activation functions that are γ-averaged, $\gamma \in (0, 1]$ (cont.)

Parametric rectified linear unit	$r_{12}(x) = \lambda x \mathbb{I}_{\{x<0\}} + x \mathbb{I}_{\{x \geq 0\}}$	1
Sigmoid linear unit (Sigmoid shrinkage)	$r_{13}(x) = \dfrac{x}{1+e^{-x}}$	1
Swish	$r_{14}(x) = \epsilon x \; sigmoid(\lambda x)$	$\dfrac{10+11\epsilon}{20}$
Gaussian	$r_{15}(x) = e^{-\langle x, x \rangle}$	$\dfrac{1+e^{-1}}{2}$
Maxout	$r_{16}(x) = \max\limits_{k} x_k$	1
Approximate Heaviside / Binary step	$r_{17}(x) = \sigma(x/\epsilon)$	$\dfrac{(1+4\epsilon)}{8\epsilon} \mathbb{I}_{\{\epsilon \geq 1/4\}}$
Multiquadratics	$r_{18}(x) = \sqrt{(x-\alpha)^2 + \lambda^2}$	1
Inverse multiquadratics	$r_{19}(x) = \dfrac{1}{\sqrt{(x-\alpha)^2 + (1+\lambda)^2}}$	$\dfrac{2+\lambda}{2(1+\lambda)}$
Mish	$r_{20}(x) = x \; tanh(\; softplus(x))$	see composition

Table: Activation functions that are γ-averaged, $\gamma \in (0, 1]$ (cont.)

Metallic mean	$r_{21}(x) = \frac{x + \sqrt{x^2 + 4}}{2}$	$\frac{1}{2}$
Arc tangent	$r_{22}(x) = tan^{-1}(x)$	1
Softsign	$r_{23}(x) = \frac{x}{1 + \lvert x \rvert}$	1
Inverse square root unit	$r_{24}(x) = \frac{x}{\sqrt{1 + (1 + \lambda)x^2}}$	$\frac{1 + \sqrt{1 + \lambda}}{2\sqrt{1 + \lambda}}$
Inverse square root linear unit	$r_{25}(x) = \frac{x}{\sqrt{1 + \lambda x^2}} \mathbb{I}_{\{x<0\}} + x\mathbb{I}_{\{x \geq 0\}}$	1
Square nonlinearity	$r_{25}(x) = -\mathbb{I}_{\{x<-2\}} + (x + \frac{x^2}{4})\mathbb{I}_{\{-2 \leq x<0\}} + (x - \frac{x^2}{4})\mathbb{I}_{\{0 \leq x \leq 2\}} + \mathbb{I}_{\{x>2\}}$	1

Table: Activation functions that are γ-averaged, $\gamma \in (0, 1]$ (cont.)

Bent identity	$r_{26}(x) = \frac{2}{3}\lambda(x + \frac{-1+\sqrt{1+x^2}}{2})$	λ
Softexponential	$r_{27}(x) = -\frac{\log(1-\lambda(x+\lambda))}{\lambda}\mathbb{I}_{\{\lambda<0\}} + x\mathbb{I}_{\{\lambda=0\}} + (\lambda + \frac{e^{\lambda x}-1}{\lambda})\mathbb{I}_{\{\lambda>0\}}$	1
Soft clipping	$r_{28}(x) = \frac{1}{\lambda}\log(\frac{1+e^{\lambda x}}{1+e^{\lambda(x-1)}})$	
	$\tilde{r}_{28}(x) = (\frac{x_k}{1+\|x\|})_k$	see r_{23}
Sinusoid	$r_{29}(x) = sin(x)$	1
Sinc	$r_{29}(x) = \frac{sin(x)}{x}\mathbb{I}_{\{x\neq 0\}} + \mathbb{I}_{\{x=0\}}$	1

Table: Activation functions that are γ-averaged, $\gamma \in (0, 1]$ (cont.)

Piecewise linear	$r_{30}(x) = 0\mathbb{I}_{\{x \leq -\frac{1}{2}\}} + (x + \frac{1}{2})\mathbb{I}_{\{-\frac{1}{2} < x < \frac{1}{2}\}} + \mathbb{I}_{\{x > \frac{1}{2}\}}$	1
Sinu-sigmoidal Linear Unit	$r_{32}(x) = (x + \lambda \sin(\alpha x))\ sigmoid(x)$	see composition
Complementary Log-Log	$r_{33}(x) = 1 - e^{-e^x}$	3/4
Bipolar Sigmoid	$r_{34}(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$	see r_5
Hard Tanh	$r_{35}(x) = \max(-1, min(1, x))$	1
Absolute value	$r_{36}(x) = \lvert x \rvert$	1
Logit	$r_{36}(x) = \frac{1}{10} \log(\frac{x}{1-x})\mathbb{I}_{[1/4,3/4]}(x)$	3/4
Softsign(Probit)	$r_{37}(x) = softsign(\Phi^{-1}(x)), \quad \Phi(x) = \frac{1}{2}\left[1 + erf(\frac{x}{\sqrt{2}})\right]$	$\frac{9}{10}$

Table: Activation functions that are γ-averaged, $\gamma \in (0, 1]$ (cont.)

Linear Gaussian	$r_{38}(x) = xe^{-x^2}$	$\frac{2+\sqrt{e}}{4}$
Attention-based	$r_{39} = softmax \circ r_0$	see composition
Attention-based	$r_{40} = r_{38} \circ softmax \circ r_0$	see composition

Main result 1: The asymptotic outcomes of DNN are Nash equilibria of a non-zero sum game

Theorem

- *The asymptotic outcomes of the deep neural network* $(x_0, r, W, b, L, \{\mathcal{H}_l\}_{1\leq l\leq L})$ *are exactly the Nash equilibria of the non-zero sum game*

$$\mathcal{G} = (\mathcal{L}, (\mathcal{H}_l,\ f_l(\cdot) + \frac{1}{2}\|\cdot - b_l - W_l x_{l-1}\|^2)_{l\in\mathcal{L}})$$

  defined by
  - *the decision-makers (players) are members of* $\mathcal{L}$
  - *the action space of the decision-maker* l *is* $\mathcal{H}_l$
  - *the objective function of decision-maker* l *output is* $z \mapsto f_l(z) + \frac{1}{2}\|z - b_l - W_l x_{l-1}\|^2$

- *For three or more layers* ($L \geq 3$)*, there is a deep neural architecture such that the resulting game is not a potential game.*

Note that this also corresponds to the (multi-level) Stackelberg solution of the game $\mathcal{G}$ as a layer reacts to the previous layer design.

Main result 2: Training and design

Given the data $(x_t, y_{L,t})_{t\in\{1,\dots,T\}}$ the training problem is to find $\theta^* = (\theta_1^*, \dots, \theta_L^*)$ such that

$$r_L \circ A_{L,\theta_{L,t}^*} \circ \dots \circ (r_1 \circ A_{1,\theta_{1,t}^*})(x_t) - y_{L,t} = 0, \quad t \in \{1, \dots, T\}, \tag{6}$$

where $\theta_l := (W_l, b_l)$ and $A_{l,t} := A_{l,(W_{l,t},b_{l,t})} : (W_{l,t}, b_{l,t}) \mapsto W_{l,t}x_{l-1,t} + b_{l,t}$ with $x_{l-1,t}$ being the output from layer $l-1$.

Theorem (A verification theorem)

Suppose that each r_k is γ_k-averaged for some $\gamma_k \in (0,1)$, $1 \le k \le L$. Then, the solutions (if any) of the training problem (6) *are also solutions of the following variational inequality:*

Given an input-output data set $\{(x_{0,t}^*, y_{L,t}^*),\ t \in \{1, 2, \dots, T\}$, find (W^*, b^*) such that

$$0 \in A_{L,t}^*[y_{L,t}^* + \partial f_L(y_{L,t}^*) - (W_{L,t}^* x_{L-1,t}^* + b_{L,t}^*)], \quad t \in \{1, 2, \dots, T\}.$$

These are also Nash equilibria of the non-zero sum game given by

$$\begin{cases} l \in \mathcal{L}, \\ \theta_l^* = (W_l^*, b_l^*) \in \arg\min_{W_l, b_l} \sum_{t=1}^{T} \omega_{l,t} \| y_{l,t}^* + \partial f_l(y_{l,t}^*) - (W_{l,t} x_{l-1,t}^* + b_{l,t}) \|^2. \end{cases} \tag{7}$$

Main result 3: Gradient descent algorithm

Theorem

Suppose the training problem has at least one solution. Then, the set of solutions of the training problem coincides with the set of Nash equilibria of the following layer by layer non-zero sum game:

$$\arg\min_{\theta_l} \sum_{t=1}^{T} \omega_{l,t}[g_l(A_{l,t}\theta_{l,t}) - \langle A_{l,t}\theta_{l,t}, y_{l,t}\rangle], \quad l \in \mathcal{L}, \;\; \omega_{l,t} > 0, \; \sum_{t=1}^{T} \omega_{l,t} = 1. \tag{8}$$

Moreover, given $\theta_0 := (\theta_{0,1}, \ldots, \theta_{0,T})$*, the algorithm*

$$\theta_{k,t}^{p+1} = \theta_{k,t}^{p} - \frac{\gamma}{2\|A_{k,t}\|^2} A_{k,t}^{*}[r_k(A_{k,t}\theta_{k,t}^{p}) - y_{k,t}], \quad 0 < \gamma < 1,$$

converges to a minimizer θ_k *of Problem* (8)*, as* $p \to \infty$*.*

Recall that

$$r_l = \partial[(\frac{1}{2}\|\cdot\|^2 + f_l)^*] := \partial g_l. \tag{9}$$

Extension

The approach extends to show the following.

- The outcomes of large language models are Nash equilibria of a non-potential game
- The outcomes of federated learning are Nash equilibria

Future work

- Extension to risk-aware NN outcomes and risk-aware training & design
- Risk-aware generative AI using mean-field-type games after appropriate random perturbations


Bauschke, H. H. and Combettes, P. L. (2011): Convex analysis and monotone operator theory in Hilbert spaces. Vol. 408. New York, Springer.


Baillon, J. B., Combettes, P. L., and Cominetti, R. (2012). There is no variational characterization of the cycles in the method of periodic projections. Journal of Functional Analysis, 262(1), 400-408.


Moreau, J. J.(1965): Proximite et dualite dans un espace hilbertien, Bull. Soc. Math. France, vol. 93, pp. 273-299.


Tembine, H.: Deep Learning Meets Game Theory: Bregman-Based Algorithms for Interactive Deep Generative Adversarial Networks, IEEE Trans Cybern, 50(3):1132- 1145, March 2020.


Shijie Wu, S., Irsoy, O., Steven Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., and David Rosenberg, D. and and Mann, G. (2023): BloombergGPT: A Large Language Model for Finance. Preprint: arXiv:2303.17564 [cs.LG].

Thank You!