Propuesta de trabajo de maestría
Machine learning methods to solve PDEs

Estudiante: Carlos Daniel Contreras Quiroz
Asesor: Mauricio Junca

Octubre 2022

1. Descripción

Las ecuaciones en derivadas parciales aparecen comúnmente como herramientas útiles para la modelación en múltiples disciplinas. Se encuentran frecuentemente aplicaciones en ciencias naturales como la física y biología, en diseño en ingeniería , y también en áreas como la economía y finanzas. Sin embargo, las propiedades matemáticas de las ecuaciones que aparecen son tan diversas como las áreas en que se aplican, y aunque se pueden clasificar parcialmente según algunas de sus características, no podría existir una teoría completa que describa nuestro conocimiento sobre estas.

Por otro lado, las soluciones analíticas a estos modelos generalmente no están a nuestro alcance, por lo que es necesario recurrir a métodos numéricos para obtener aproximaciones. Para esto, usualmente se recurre a métodos clásicos como diferencias finitas, elementos finitos, volúmenes finitos o métodos espectrales, para los cuales existe una amplia teoría que soporta y justifica rigurosamente su funcionamiento.

No obstante, la aplicación de estos métodos a problemas particulares a veces se restringe por propiedades especificas de la ecuación que se resuelve. Por ejemplo, los métodos mencionados sufren de la maldición de la dimensionalidad (*"the curse of dimentionality"*), esto es, su complejidad computacional escala exponencialmente en la dimensión del problema, por lo que su uso se restringe a problemas de dimensión baja ($n = 1, 2, 3, 4$). Lo anterior dificulta su implementación en aplicaciones como valoración en matemática financiera, donde la dimensión del problema está determinada por el número de activos considerados . También, su eficiencia computacional se reduce considerablemente conforme se aumenta la complejidad de los dominios en que se resuelven, o por las no-linealidades que aparecen, como es el caso de la ecuación de Navier-Stokes modelando flujos turbulentos.

Otra área en donde estos inconvenientes aparecen es en el análisis de datos y aprendizaje de maquinas. Por ejemplo, la complejidad de algunos modelos de regresión no lineal crece exponencialmente con el tamaño de los datos subyacentes. Para este tipo de problemas se han desarrollado herramientas poderosas que permiten modelar problemas en altas dimensiones y con posibles no linealidades. Entre estas, las redes neuronales han demostrado ser de gran utilidad como modelo para representar funciones con estas complejidades[1].

En consecuencia, intentando replicar el éxito obtenido con estas herramientas en aprendizaje de máquinas, recientemente han surgido nuevas perspectivas para aproximar soluciones de ecuaciones en derivadas parciales usando estas mismas herramientas. Entre estas se encuentran las PINNs (Physics Informed Neural Networks)[2, 3], FNO (Fourier Neural Operators)[4], y DGM (Deep Garlekin Method)[5]. La evidencia práctica muestra que estos métodos pueden proporcionar soluciones en casos donde los clásicos no [6, 7], a pesar de usualmente no competir con su eficiencia en las situaciones donde los últimos sí aplican. Además, se ha venido desarrollando un marco teórico riguroso que permite justificar su aplicación en situaciones específicas.

Como objetivo razonable de este trabajo se propone estudiar teóricamente e implementar computacionalmente algunos de estos métodos, buscando establecer sus ventajas y alcances. La cantidad de métodos a estudiar dependerá de las dificultades encontradas y las restricciones de tiempo que se deben cumplir. A continuación se mencionan los métodos elegidos inicialmente.

El primero de estos se propone en [8], y se basa en explotar la conexión que existe entre algunas ecuaciones parabólicas (posiblemente no lineales) y las ecuaciones diferenciales estocásticas hacia atrás (BSDE Backward Stochastic Differential Equations)[9]. En este caso se considera ecuaciones parabólicas semilineales de la forma

$$\begin{aligned}\frac{\partial u}{\partial t}(t,x)+\frac{1}{2}\operatorname{Tr}\left(\sigma\sigma^{\mathrm{T}}(t,x)\left(\operatorname{Hess}_x u\right)(t,x)\right)+\nabla u(t,x)\cdot\mu(t,x)\\ +f\left(t,x,u(t,x),\sigma^{\mathrm{T}}(t,x)\nabla u(t,x)\right)=0,\end{aligned} \tag{1}$$

con condición final $u(T,x) = g(x)$, donde t representa el tiempo, x la variable espacial d-dimensional, μ una función vectorial conocida, σ una matriz $d \times d$ y f una función no lineal. Similar al caso lineal, en donde a la solución a la ecuación diferencial parcial parabólica se le puede asignar una representación en términos del valor esperado de un proceso asociado a una ecuación diferencial estocástica a través de la formula de Feynman-Kac, se puede asignar una representación de la solución de la ecuación 1 en términos de la solución de una BSDE de la forma

$$\begin{aligned}&u\left(t,X_t\right)-u\left(0,X_0\right)\\ =&-\int_0^t f\left(s,X_s,u\left(s,X_s\right),\sigma^{\mathrm{T}}\left(s,X_s\right)\nabla u\left(s,X_s\right)\right)ds\\ &+\int_0^t\left[\nabla u\left(s,X_s\right)\right]^{\mathrm{T}}\sigma\left(s,X_s\right)dW_s,\end{aligned} \tag{2}$$

donde X_t es la solución de la SDE

$$X_t=\xi+\int_0^t\mu\left(s,X_s\right)ds+\int_0^t\sigma\left(s,X_s\right)dW_s. \tag{3}$$

En cada paso de tiempo, la ecuación 3 se discretiza mediante el método de Euler-Maruyama para obtener muestras de caminos $\{X_n\}_{0\le n\le N}$, y para usar la representación de la solución 2 se aproxima la función $x \to \sigma^T(t,x)\nabla u(t,x)$ con una red neuronal que se entrena minimizando la función de costo

$$l(\theta)=\mathbb{E}\left[\left|g\left(X_{t_N}\right)-\hat{u}\left(\left\{X_{t_n}\right\}_{0\le n\le N},\left\{W_{t_n}\right\}_{0\le n\le N}\right)\right|^2\right], \tag{4}$$

relacionada con el cumplimiento de las condiciones iniciales de la ecuación. Al final, el conjunto de redes neuronales optimizadas proporciona la aproximación de la solución requerida.

En este artículo se presenta la aplicación de este método para resolver un modelo de Black-Scholes, una ecuación de HJB para un problema de control y una ecuación de Allen-Cahn, todas en dimensiones altas (≥ 10). A parte de estudiar la relación entre ecuaciones parabólicas no lineales y BSDEs y su uso en este métodos, se pretende revisar la literatura que justifica rigurosamente su uso[10, 11, 12] e implementar uno de los ejemplos dados para estudiar la eficiencia y propiedades prácticas de la solución.

El segundo método que se quiere estudiar, en caso que el tiempo lo permita, se propone en [13], y se inspira en el algoritmo de Q-learning de reinforcement learning. En este caso se estudian problemas más generales de la forma

$$\begin{aligned} \mathcal{L}u &= 0 \text{ en } \Omega \\ u &= f \text{ en } \partial\Omega, \end{aligned} \tag{5}$$

donde $\mathcal{L}$ es un operador no lineal de segundo orden con ciertas condiciones de regularidad, y Ω es un dominio. Aquí se quiere usar una red neuronal de una capa con N neuronas de la forma

$$S^N(x;\theta) = \frac{1}{N^\beta} \sum_{i=1}^{N} c^i \sigma\left(w^i \cdot x + b^i\right), \tag{6}$$

donde w^i, b^i, c^i son los parámetros de la red y β es un factor de escalamiento. Con esta se propone una aproximación de la función

$$Q^N(x;\theta) := S^N(x;\theta)\eta(x) + (1 - \eta(x))\bar{f}(x), \tag{7}$$

donde $\eta(x)$ es una función suave con $0 < \eta(x) < 1$ en el interior de Ω y $\eta(x) = 0$ en $\partial\Omega$ y $\bar{f}$ es tal que $T(\bar{f}) = f$, con $T : \mathcal{H}^1 \to L^2(\partial\Omega)$ el operador de traza sobre la frontera. Esta red se entrenará en cada paso intentando minimizar la función de costo

$$\int_\Omega \left[\mathcal{L}Q^N(x)\right]^2 d\mu(x), \tag{8}$$

la cual se puede estimar por métodos de Monte-Carlo. Al final Q^N proporcionará una aproximación de la solución que cumple las condiciones de frontera impuestas.

De la misma manera, se buscará entender el soporte teórico proporcionando una prueba de convergencia para este algoritmo y posteriormente implementarlo para evaluar su utilidad práctica.

2. Objetivos

- Estudiar la relación entre BSDE y ecuaciones parabólicas no lineales.
- Explotar esta relación para resolver una ecuación diferencial no parabólica usando métodos de Deep Learning.
- Estudiar el algoritmo Q-PDE, basado en Q-learning, para resolver ecuaciones diferenciales elípticas.
- Implementar computacionalmente estos métodos para evaluar su utilidad práctica.

3. Cronograma tentativo

- Septiembre-Octubre 2022: Aprender a implementar redes neuronales como métodos de aproximación para funciones Q
- Noviembre-Diciembre 2022: Estudiar la relación entre BSDE y ecuaciones diferenciales parabólicas no lineales.
- Enero 202: Implementar método basado en BSDE.
- Febrero-Marzo 2023: Estudiar e implementar el algoritmo Q-PDE
- Abril 2023: Terminar el documento

Referencias

[1] C. F. Higham and D. J. Higham, "Deep Learning: An Introduction for Applied Mathematicians," *SIAM Review*, vol. 61, pp. 860–891, Jan. 2019.

[2] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics informed deep learning (part I): data-driven solutions of nonlinear partial differential equations," *CoRR*, vol. abs/1711.10561, 2017.

[3] S. Markidis, "The old and the new: Can physics-informed deep-learning replace traditional linear solvers?," 2021.

[4] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier Neural Operator for Parametric Partial Differential Equations," May 2021. arXiv:2010.08895 [cs, math].

[5] J. Sirignano and K. Spiliopoulos, "DGM: A deep learning algorithm for solving partial differential equations," *Journal of Computational Physics*, vol. 375, pp. 1339–1364, Dec. 2018.

[6] S. Cuomo, V. S. di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next," June 2022. arXiv:2201.05624 [physics].

[7] J. Blechschmidt and O. G. Ernst, "Three ways to solve partial differential equations with neural networks — A review," *GAMM-Mitteilungen*, vol. 44, June 2021.

[8] J. Han, A. Jentzen, and W. E, "Solving high-dimensional partial differential equations using deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, pp. 8505–8510, Aug. 2018.

[9] D. Revuz and M. Yor, *Continuous Martingales and Brownian Motion*. Springer Berlin Heidelberg, 1999.

[10] W. E, J. Han, and A. Jentzen, "Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations," *Communications in Mathematics and Statistics*, vol. 5, pp. 349–380, nov 2017.

[11] C. Beck, W. E, and A. Jentzen, "Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations," *Journal of Nonlinear Science*, vol. 29, pp. 1563–1619, jan 2019.

[12] M. Hutzenthaler, A. Jentzen, T. Kruse, and T. A. Nguyen, "A proof that rectified deep neural networks overcome the curse of dimensionality in the numerical approximation of semilinear heat equations," *SN Partial Differential Equations and Applications*, vol. 1, Apr. 2020.

[13] S. N. Cohen, D. Jiang, and J. Sirignano, "Neural Q-learning for solving elliptic PDEs," p. 57, 2022.

[14] C. Beck, M. Hutzenthaler, A. Jentzen, and B. Kuckuck, "An overview on deep learning-based approximation methods for partial differential equations," Mar. 2021. arXiv:2012.12348 [cs, math].

[15] L. Lu, P. Jin, and G. E. Karniadakis, "DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators," *Nature Machine Intelligence*, vol. 3, pp. 218–229, Mar. 2021. arXiv:1910.03193 [cs, stat].

[16] S. Mishra, "A machine learning framework for data driven acceleration of computations of differential equations," July 2018. arXiv:1807.09519 [cs, math].

[17] C. Beck, M. Hutzenthaler, A. Jentzen, and B. Kuckuck, "An overview on deep learning-based approximation methods for partial differential equations," Mar. 2021. arXiv:2012.12348 [cs, math].

[18] J. B. Pedro, J. Maroñas, and R. Paredes, "Solving Partial Differential Equations with Neural Networks," Dec. 2019. arXiv:1912.04737 [physics].