Machine Learning vs. Deep Learning: A Comparison Through Two Real-World Applications

Manas Maheshwari

Indiana Wesleyan University

AIML-500: Machine Learning Fundamentals

Dr. Eskinder Belete

July 9, 2026

Machine Learning vs. Deep Learning: A Comparison Through Two Real-World Applications

Machine learning and deep learning are often discussed as if they were competing technologies, but in practice they are tools suited to different kinds of problems. Traditional machine learning relies on humans to select the features that matter, then applies algorithms such as support vector machines or logistic regression to learn from those features. Deep learning uses multi-layer neural networks that discover the important features on their own directly from raw data. The practical question for any organization is therefore not which approach is more advanced, but which approach matches the structure of the data and the constraints of the problem. This report examines one example of each from the lesson: customer churn prediction using a support vector machine as the traditional machine learning case, and natural language processing with Transformer models as the deep learning case. For each, it identifies a real-world application, explains why the chosen approach fits the problem, and explains why the opposite approach would be a poor fit.

Example 1: Customer Churn Prediction with a Support Vector Machine

Customer churn prediction asks a simple business question: which current customers are likely to leave? The lesson presents this as a traditional machine learning problem in which a company uses features such as customer age, contract length, and monthly charges, and a support vector machine (SVM) learns from historical records of customers who stayed or left (Cortes & Vapnik, 1995). A real-world application is subscription retention in the telecommunications industry, where carriers score their subscriber base each month so that retention teams can intervene, for example with a targeted offer, before a customer cancels. The same pattern appears throughout subscription businesses, including streaming services and financial services firms monitoring account attrition.

Traditional machine learning is well suited to this problem for three reasons. First, the data is structured and tabular. Each customer is already described by a fixed set of meaningful columns: tenure, plan type, monthly charges, support call frequency. Feature engineering, the

supposed weakness of traditional machine learning, is nearly free here, because human experts already know which attributes plausibly drive cancellation and those attributes already exist as database fields. Second, the datasets are modest in size. Even a large carrier has millions of customers, not billions of examples, and traditional algorithms such as SVMs perform strongly at this scale while training quickly and cheaply. Third, churn models drive business decisions about real people, so stakeholders need to understand and audit why the model flagged a customer. Simpler models over human-chosen features are far easier to explain to a retention manager, an executive, or a regulator.

Deep learning is not suitable here, and the reasons mirror the strengths above. A deep neural network earns its complexity by discovering features automatically from raw, high-dimensional data such as pixels or free text. Churn data offers no such opportunity: the features are already explicit columns, so there is little hidden structure for the network to discover, and research on tabular data repeatedly finds that deep models fail to outperform simpler ones in this setting. Meanwhile, deep learning would impose real costs. It typically requires far more data and computing power to reach its potential, it is slower and more expensive to train and tune, and its layered representations make its individual predictions difficult to explain. Applying deep learning to churn prediction would add cost and opacity without adding accuracy, which is the definition of an unsuitable tool.

Example 2: Natural Language Processing with Transformer Models

The deep learning example is natural language processing using Transformer models (Vaswani et al., 2017). The real-world application is machine translation. Google Translate moved from a phrase-based statistical system to a neural machine translation system built on deep networks, reporting large reductions in translation errors, and modern translation systems are built on Transformer architectures that process a sentence as a whole and learn the relationships among its words (Wu et al., 2016). Hundreds of millions of people now rely on this capability daily across the web, mobile devices, and customer support workflows.

Deep learning is suitable for translation because the problem is exactly the one neural networks were built for: the input is raw, unstructured, sequential data whose meaning depends on context. The correct translation of a word depends on the words around it, on grammar, on idiom, and on long-range relationships across the sentence. No human can write down a complete feature checklist that captures these relationships, but a deep network can learn them, layer by layer, from enormous volumes of parallel text. The scale requirements that made deep learning wrong for churn are satisfied here: translation systems train on hundreds of millions of sentence pairs, and the economic value of the application justifies the computational cost. Deep learning also handles the output side, since translation requires generating a fluent sequence, not just assigning a label, and sequence-to-sequence neural architectures are designed to do precisely that.

Traditional machine learning is not suitable for this problem. A feature-engineered approach would require humans to manually define the characteristics of language that determine meaning, such as word order, agreement, ambiguity, and idiom, across every pair of languages. Decades of pre-neural machine translation effort demonstrated how brittle this is: systems built on hand-crafted rules and simple statistical features produced translations that were often literal, ungrammatical, or wrong, because fixed features cannot capture context. There is also a structural mismatch. Algorithms such as SVMs or logistic regression predict a single label or number, but translation must produce an ordered sequence of words of varying length. The traditional toolkit does not merely perform worse on this task; it is not shaped to perform the task at all.

Conclusion

Placed side by side, the two examples make the dividing line concrete. When data is structured, features are already known and meaningful, datasets are modest, and decisions must be explainable, traditional machine learning is the right tool, as in churn prediction. When data is raw and unstructured, the important features are beyond human ability to specify, massive training data is available, and the output itself is complex, deep learning is not just preferable but

necessary, as in machine translation. The lesson's summary analogy captures it well: machine learning hands the student a checklist, while deep learning teaches the student to read. The professional skill is recognizing which kind of problem is on the desk before choosing the student.

References

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273-297.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., . . . Dean, J. (2016). *Google's neural machine translation system: Bridging the gap between human and machine translation*. arXiv. https://arxiv.org/abs/1609.08144