2021-01-05 Notes TX-RX (FL)

**BIAS**

1. Types of machine learning bias

There are various ways that bias can be brought into a machine learning system. Common scenarios, or types of bias, include the following:

* **Algorithm bias.** This occurs when there's a problem within the [algorithm that performs the calculations](https://searchenterpriseai.techtarget.com/feature/5-types-of-machine-learning-algorithms-you-should-know) that power the machine learning computations.
* **Sample bias.** This happens when there's a problem with the data used to train the machine learning model. In this type of bias, the data used is either not large enough or representative enough to teach the system. For example, using training data that features only female teachers will train the system to conclude that all teachers are female.
* **Prejudice bias.** In this case, the data used to train the system reflects existing prejudices, stereotypes and/or faulty societal assumptions, thereby introducing those same real-world biases into the machine learning itself. For example, using data about medical professionals that includes only female nurses and male doctors would thereby perpetuate a [real-world gender stereotype](https://searchhrsoftware.techtarget.com/feature/Hiring-vendor-says-gender-based-AI-bias-is-pervasive) about healthcare workers in the computer system.
* **Measurement bias.** As the name suggests, this bias arises due to underlying problems with the accuracy of the data and how it was measured or assessed. Using pictures of happy workers to train a system meant to assess a workplace environment could be biased if the workers in the pictures knew they were being measured for happiness; a system being trained to precisely assess weight will be biased if the weights contained in the training data were consistently rounded up.
* **Exclusion bias.** This happens when an important data point is left out of the data being used --something that can happen if the modelers don't recognize the data point as consequential.

1. Bias vs. variance

Data scientists and others involved in building, training and using machine learning models must consider not just bias, but also variance when seeking to create systems that can deliver consistently accurate results.

Like bias, variance is an error that results when the machine learning produces the wrong assumptions based on the training data. Unlike bias, variance is a reaction to real and legitimate fluctuations in the data sets. These fluctuations, or noise, however, should not have an impact on the intended model, yet the system is using that noise for modeling. In other words, variance is a problematic sensitivity to small fluctuations in the training set, which, like bias, can produce inaccurate results.

Although bias and variance are different, they are interrelated in that a level of variance can help reduce bias. If the data population has enough variety in it, biases should be drowned out by the variance.

As such, the objective in machine learning is to have a tradeoff, or balance, between the two in order to develop a system that produces a minimal amount of errors.

1. How to prevent bias

Awareness and good governance can help prevent machine learning bias; an organization that recognizes the potential for bias can then implement and institute [best practices to combat it](https://searchenterpriseai.techtarget.com/feature/6-ways-to-reduce-different-types-of-bias-in-machine-learning) that include the following steps:

1. Select training data that is appropriately representative and large enough to counteract common types of machine learning bias, such as sample bias and prejudice bias.
2. Test and validate to ensure the results of machine learning systems don't reflect bias due to algorithms or the data sets.
3. Monitor machine learning systems as they perform their tasks to ensure biases don't creep in over time as the systems continue to learn as they work.
4. Use additional resources, such as Google's [What-if Tool](https://pair-code.github.io/what-if-tool/) or IBM's [AI Fairness 360 Open Source Toolkit](https://aif360.mybluemix.net), to examine and inspect models.

**POSSIBILI TECNICHE DI TX/RX**

1. Codifica Manchester:
   1. Specifica che per un bit 0 i livelli di segnale saranno Basso-Alto (assumendo una codifica dei dati con l'ampiezza) - con un livello basso nella prima parte del periodo di bit, ed un livello alto nella seconda parte. Per un bit 1 i livelli di segnale saranno Alto-Basso.
   2. La codifica Manchester si può considerare come un caso speciale della [Phase Shift Keying Binaria](https://it.wikipedia.org/wiki/BPSK) (BPSK), in cui il dato da trasmettere controlla la [fase](https://it.wikipedia.org/wiki/Fase_(segnali)) di un'onda quadra [portante](https://it.wikipedia.org/wiki/Portante) alla frequenza della velocità di trasmissione dati. Perciò è estremamente facile generare tale segnale in modo digitale.
2. Modulazione di ampiezza (digitale - analogica):
   1. Valutazione dell'inviluppo per identificare la tx del bit "1" o "0";
      1. <https://stackoverflow.com/questions/34235530/python-how-to-get-high-and-low-envelope-of-a-signal>
      2. <https://stackoverflow.com/questions/30889748/how-to-obtain-sound-envelope-using-python>
3. Tecniche di codifica/quantizzazione dei segnali:
   1. Differential Pulse Code Modulation;
   2. Adaptive Delta Modulation.

**CAPACITA' DI CANALE:**

1. Massimo numero di bit/s = Banda del canale senza rumore \* LOG2 (1 + S/N)
   1. Da capire come definire S ed N in FL.

**TX/RX:**

FIGURA rispetto al modello GLOBALE

A picture containing diagram

Description automatically generated

* Canale diviso in slot:
  + TX = SEND
  + RX
  + RESET

* SLOT:
  + Sono definiti in termini di clicli di addestramento, quindi sono tempo-discreti.

* Step 1 (SETUP LARGHEZZA SLOT)
  + Il sender utilizza un segnale/digit di controllo con cui all'inizio effettua delle misurazioni per capire la lunghezza in termini di cicli (EPOCHE) di addestramento è lungo ogni slot.

* Step 2 (Sincronismo TX-RX)
  + Serve pensare ad un meccanismo di sincronismo tramite un codice binario per cui Sender e Receiver capiscono che possono iniziare la comunicazione.
  + Oppure un protocollo tipo Three-Way-Handshake (<https://www.guru99.com/tcp-3-way-handshake.html>)
* Step 3 (Comunicazione)
  + Trasmettiamoi dati secondo uno schema:
    - Modulazione di ampiezza
    - Modulazione di fase (se troviamo un modo per far diminuire il BIAS/SCORE in modo artificiale)

**NOTE:**

* Capacita del canale:
  + Possiamo usare la formula:
    - Massimo numero di bit/s = Banda del canale senza rumore \* LOG2 (1 + S/N)
    - S = SEGNALE --> SENDER
    - N = RUMORE --> TUTTI GLI ALTRI CHE TX
* Canale adattativo al modello di FL:
  + Più saranno ripide le curve con cui AUMENTA/DIMINUISCE lo score/bias e più saranno stretti gli SLOT, quindi si potranno trasmettere molti più bit per secondo.