

YASH BONDE

CONTACT

Machine Learning Engineer

Ardent machine learning practitioner and hobby researcher, normally found experimenting. If not doing that probably reading a book or cooking. I want to work towards solving fundamental scientific research problems using Artificial Intelligence and Computation.

+91-91310-83106

bonde.yash97@gmail.com

github.com/yashbonde

yashbonde.github.io

linkedin.com/in/yash-bonde

Raipur, India

WORK EXPERIENCE

Machine Learning Engineer

Logistixian Technologies, Ahmedabad (July 2019 - Present)

- Convert unstructured business data (documents) to insights.
- Single handedly researched and implemented novel deep learning models.
- Managing operations from requirements gathering to deployment for our clients, some of the biggest shipping lines on the globe.

Freelance ML Engineer

- Custom Language Models (2019)
- High volume video analysis tool on MS Azure (2018)
- Audio segmentation using custom DL model (2018)

Summer Internship, Computer Vision

Kaanaat, Bengaluru (May 2018 - October 2018)

- Model for auto-determining traffic flow and perform anomaly detection on-the-edge.
- Developed application for faster image segmentation. Helped ADAS team on image stabilisation, orientation and object tracking.

Summer Internship, Research Intern

Connecticus Technologies, Pune (May 2017 - July 2017)

- Implemented research papers to make a FAQ bot.

Summer Internship, Software

Mastersoft E.R.P. Solutions, Nagpur (May 2016 - July 2016)

- Built a data visualisation toolkit for ERP systems.

EDUCATION

National Institute of Technology, Raipur (2015 - 2019)

Bachelors of Technology, Electronics and Telecommunications

- Part of Training and Placement Office team that managed the annual placements operations for institute
- Active in various clubs and cells. Headed Tech. and Design teams for The Entrepreneurship Cell
- Country finalist in MSAIC 2018, Country quarter-finalist in TIIDC 2017 for building a prototype of a wearable Indian Sign Language Converter.

PERSONAL PROJECTS & RESEARCH*

* More details on following page

The projects worked on can be summarised as follows:

Personal Research:

- Fundamental Mathematics using AI
- Neural Algorithmic Execution
- Weather Modelling using Recurrent GNNs

For fun projects:

- Game Development using GANs
- Applied Reinforcement Learning
- Drug Multitask classification

Implementation: Several research papers for practice

Software Dev.: Projects on which spent considerable effort on software engineering part such as a fullstack Chess App

SKILLS

- Fast prototyping to implementation
- Can manage pilot projects and UATs
- Research driven approach

INTERESTS

- AI for social good and better governance
- Better learning tools to make re-skilling easier
- Accelerate rate of progress in core sciences using AI

PRIMARY TECH STACK

- **Languages:** python, JS, SQL, shell
- **Tools:** Django, FastAPI, Pytorch, Tensorflow

DETAILS ON SOLO PROJECTS & RESEARCH (R)

GAN Games (2020-) [Repo]

Can we use GANs to generate game components like levels, textures, play styles and combine them to form a new game every single time. Often the procedurally generated content is very limiting in its possibilities due to the hand engineered features. GANs are perfectly suited to this task as they can be tuned to features while still being a random generator.

(R) Fundamental Mathematics Research using AI

Experimenting first known solution of scientific problem. If PoC is successful, it would drastically reduce time between observation to theory conversion and vice-versa in physics, chemistry and other sciences with mathematics at core.

(R) Weather Modelling using Recurrent GNNs: (2020-) [Code]

Weather is a global and it is also a time based phenomenon moreover one event can have an effect on somewhere far off. A Graph Network that act as encoder-decoder for a big LSTM can capture this. LSTMs because of their recurrence property and long term memory can be assumed to model complicated phenomenon. Trained on South Brazil Weather dataset.

(R) Freeciv Learning Environment (2019-2020) [Repo]

Tried converting popular open source turn-based strategy game into a learning environment to train RL agents. Build preliminary bindings for python and experiment with training model trained on action replays using supervised learning. Could not complete the project due to technical challenges and computational requirements for running the models.

Reinforcement Learning Study and Practice [Blogs]

Detailed notes on Barto & Sutton's "Introduction to Reinforcement Learning", solved the exercises given in the book. Has helped many people across the world by providing easy to understand explanations. Experimented with variety of methods to use RL for knowledge structuring but ended up with better supervised methods.

Applied Reinforcement Learning

Implemented different algorithms like Q-learning based (DQN, DDQN), Actor Critic (AC, A2C, A3C) and Proximal Policy Optimisation (PPO) on variety of environments like VizDoom, OpenAI gym and Freeciv Learning Environment.

Built a simple experiment to play *Satti Lavni* or card game called Sevens. Implemented my own versions of latest methods like self-play to generate more samples and MCTS for searching in the step tree.

Drug Multitask Classification (2020) [Script] and [Blog]

Graph Neural Networks are the most abstract networks, that can take in any form of data and generalise to everything. With this in mind, I wanted to learn more about them. So I implemented a simple multi-task drug classification network. Molecules were input as a graph data structure and network's job was to predict whether it was carcinogenic or not on males and females.

(R) Neural Algorithmic Execution (2020) [Blog] and [Blog]

Was it possible for neural networks to imitate conventional software algorithm. If we can implement conventional algorithms then we can completely move over to the software 2.0 stack. Implemented Conway's game of life and observed that neural networks could learn it, proving that machine learning stack is the correct evolution. After this I performed exhaustive study on the topic as there was nothing out there on the internet, so I wrote detailed notes on the topic.

Fullstack Chess App (2019 -2020) [App] and [Model]

Simply making AI models does not make them good to use, so making front-end and backend systems is crucial to make them useful. Full-stack software project with backend, frontend and AI part of a chess playing bot. Users were authenticated and could play with the bot with each move being logged. Implemented the AlphaZero architecture and partial implementation of MCTS. However training is not possible on local machine, so had to make simpler and something more practical.

Trick was to train the bot like an unsupervised (instead of RL) language model on pre-existing corpus of games. Trained the model could play chess on a decent level. Further improvements include better searching, combining MCTS with token probability.

Implemented Research Papers

- **(R) Deep Learning for Symbolic Mathematics** (Lample and Charton, 2019): Authors created a method to generate >1e38 mathematical expressions, I implemented a simpler trimmed down version of this code for personal research.
- **Language Models are Unsupervised Multitask Learners** (Radford et al., 2017): OpenAI GPT-2 paper, trained a language model on NIPS papers till 2019 as a technology demonstrator at Logistixian Technologies.
- **Attention is All You Need** (Vaswani et al., 2017): Implemented encoder-decoder Transformers for NMT, tested on English-French / Hindi / Bulgarian. Demonstrated that this method can be used not just for languages but any input that can be converted to embedding.
- **Hybrid computing using a neural network with dynamic external memory** (Graves et al., 2016): Deepmind paper introducing Differentiable Neural Computer (DNC). They showed how a neural network can be used in place of conventional computer parts like read, write heads and capabilities include long term remembrance, solving graph problems.
- **Generative Adversarial Nets** (Goodfellow et al., 2014): Implemented for MNIST dataset.
- **Conditional Generative Adversarial Nets** (Mirza et al., 2014): Trained on MNIST, Generates user specified digits instead of the random ones generated by Vanilla GAN.
- **End-To-End Memory Networks** (Sukhbaatar et al., 2015): Authors implemented a network with dedicated memory matrix that could help in solving problems that needed long term dependency.

Others

- **Text2SQL**: One high value business problem is that data owners cannot access their own databases because of lack of technical skills like writing SQL. Trained a language model to convert Natural Language text to SQL query!
- Chatbots using Seq2Seq
- **Wallstreet**: A college project where players competed with each other and against computer to earn maximum money on virtual stock market