

Redes Neurais Convolucionais Otimizadas para a Detecção de Supernovas

Gabriela Cardoso Montes, Lucas Grassano Lattari, Alessandra Martins Coelho

¹Departamento Acadêmico de Ciência da Computação - Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais - Campus Rio Pomba
Caixa Postal 45 – CEP: 36180-000 – Rio Pomba – MG – Brasil

lucas.lattari@ifsudestemg.edu.br

Abstract. *Supernovae are superluminous explosions that characterize the conclusion of a supermassive star life cycle. Artificial Neural Networks automate the analysis of supernova data from telescopes while requiring a significant amount of hardware resources for their implementation. In this paper, we propose a Pruned Convolutional Neural Network optimized for the Detection of Supernovae in images, adopting the pruning strategy on Artificial Neural Networks. The algorithm obtained an accuracy of 0.964 and the pruning was able to reduce the size of the model by up to 70%, without significant loss of accuracy.*

Resumo. *Supernovas são explosões super luminosas que caracterizam o fim da vida de uma estrela supermassiva. Redes Neurais Artificiais são ferramentas promissoras para a automação da análise dos dados de supernovas provenientes de telescópios, porém, apresentam uma desvantagem, que é a grande demanda por recursos de hardware necessários para sua implementação. Neste trabalho, propõe-se uma Rede Neural Convolutiva otimizada para Detecção de Supernovas em imagens, utilizando a técnica de poda dos parâmetros. O algoritmo obteve uma acurácia de 0,964 e a poda foi capaz de reduzir o tamanho do modelo em até 70%, sem perda significativa de acurácia.*

1. Introdução

Supernovas (SNs) caracterizam o fim da vida de estrelas muito massivas [Sobrinho 2013]. Essas explosões são eventos que envolvem muita energia e, no momento de maior brilho, podem chegar a ofuscar o brilho da própria galáxia hospedeira da estrela que originou a supernova [Burrows and Goshy 1993]. Elas podem ser divididas em vários tipos, sendo os principais: SN do tipo Ia, Ib, Ic e SN do tipo II [Sobrinho 2013]. SNs do tipo Ia, além de muito comuns no universo, são usadas para medir distâncias, conceito fundamental para a Cosmologia [Burrows and Vartanyan 2021]. Também foi por meio do estudo de SNs do tipo Ia que cientistas obtiveram a primeira evidência de que o Universo está se expandindo de forma acelerada [Perlmutter et al. 1999, Riess et al. 1998], sendo a busca pelo componente que causa a aceleração da expansão do Universo, denominado energia escura, um dos principais problemas em aberto da Física deste século [Ishida 2019].

Redes Neurais Artificiais (RNAs) são ferramentas promissoras para a automação da análise dos dados provenientes de telescópios. Porém, apesar de serem poderosas e eficientes ferramentas para a Astronomia, as RNAs apresentam uma desvantagem, que é

a grande demanda por recursos de *hardware* necessários para sua implementação. Diante disso, pesquisadores vêm buscando técnicas de compressão de modelos para reduzir a intensa demanda computacional, seja de processamento e/ou memória, sem que isso implique em uma piora dos resultados. Para que isso seja feito, uma alternativa é a adoção da poda de RNAs, proposta por [LeCun et al. 1989], que possibilita a otimização dos modelos sem perda (ou perda mínima) de precisão.

Dentre as maneiras de otimizar a estrutura de uma RNA, a poda de parâmetros é uma das técnicas mais popularmente utilizadas. A ideia principal dessa abordagem é que existe um grande número de parâmetros sem importância e redundantes, que poderiam ser eliminados (ou seja, a rede é mais complexa do que o necessário) e que, portanto, é possível simplificá-la mantendo algo equivalente ao seu desempenho original. A técnica consiste em atribuir o valor zero a parâmetros cujos valores já são muito próximos a zero, o que resulta em economia de memória e de processamento, visto que torna os cálculos mais facilitados.

Dessa forma, propõe-se neste trabalho uma arquitetura de Redes Neurais Profundas para a identificação de SNs e não SNs em imagens de telescópios. Esse modelo baseia-se em uma Rede Neural Convolutiva (CNN) otimizada por meio da poda. Almeja-se construir um modelo que apresente resultados competitivos com os demais da literatura e seja capaz de ser executado em dispositivos com poucos recursos de *hardware*, popularizando, assim, algoritmos do gênero para diferentes arquiteturas computacionais.

2. Revisão da Literatura

[Goldstein et al. 2015] descreve o Autoscan, um algoritmo cujo objetivo é classificar SNs ou não SNs em imagens de diferença (que revelam mudanças em uma região do céu em dois momentos diferentes). Foi desenvolvido para o *Dark Energy Survey*, um projeto que visa detectar SNs e encontrar padrões de estrutura cósmica para revelar a natureza da energia escura que está acelerando a expansão do Universo [Flaugher 2005]. O algoritmo de Floresta Aleatória (*Random Forest*) foi treinado com uma base de dados contendo 898.963 imagens de SNs e não SNs, disponível para *download*¹ no site do projeto e seu código-fonte pode ser encontrado no GitHub². O conjunto é composto por imagens reais e sintéticas do projeto *Dark Energy Survey*, nas quais a luz de SNs foi sobreposta por figuras de galáxias. Com o Autoscan, os pesquisadores conseguiram reduzir para um fator de 13,4 o número de imagens que os astrônomos precisam analisar. No entanto, o Autoscan possui a desvantagem de requerer a utilização de uma avançada Engenharia de Atributos para extrair as características das imagens da base de dados, sendo esta uma das etapas mais demoradas na execução do algoritmo [Goldstein et al. 2015].

Visando aprimorar o processo de identificação de supernovas proposto pelo Autoscan, um grupo de engenheiros do projeto Space2Vec, do DESSA (um laboratório de pesquisa e desenvolvimento em Inteligência Artificial) desenvolveu um algoritmo capaz de identificar SNs sem depender da mesma Engenharia de Atributos, utilizando a mesma base de dados que o Autoscan [Dessa 2019]. Para isso, eles implementaram uma RNA baseada em CNN, com nove camadas de convolução e o emprego da função de ativação

¹<https://portal.nersc.gov/project/dessn/autoscan/>

²<https://github.com/dessn/autoscan>

ReLU entre elas, uma camada densa e a camada de saída. Após o treinamento, o modelo obteve uma acurácia de 0,924. O código-fonte também está disponível no GitHub³.

Finalmente, [Cabrera-Vives et al. 2017] apresenta o algoritmo Deep-HiTS, uma CNN para detecção de SNs em imagens de diferença, especialmente em suas primeiras horas de explosão [Förster et al. 2016]. Ela é desenvolvida para o projeto *High Cadence Transient Survey* (HiTS). O modelo foi treinado com uma base de dados contendo 1.200.000 imagens, composta por não SNs obtidas do HiTS e SNs simuladas. A arquitetura do modelo conta com cinco camadas de convolução, três camadas densas e a camada de saída e obteve uma acurácia de 0,994. Seu código-fonte também está disponibilizado no GitHub⁴.

3. Materiais e Métodos

Para descrever a metodologia deste trabalho, é apresentada na Seção 3.1 a base de dados utilizada. A Seção 3.2 delimita a arquitetura da RNA, bem como os recursos de *hardware* e *software* utilizados no desenvolvimento. A Seção 3.3 explica o passo a passo do processo de otimização do modelo.

3.1. Tratamento da Base de Dados

Para analisar eventos astronômicos, podemos usar imagens de diferença. Essas imagens revelam mudanças na aparência do céu em dois períodos diferentes e são produzidas pela subtração da imagem base pela imagem do mesmo local após o evento. Para o treinamento do modelo, foi utilizada a mesma base de dados apresentada por [Goldstein et al. 2015]. A resolução espacial das imagens (altura e largura) é de 51x51 *pixels*. Das 898.963 imagens contidas na base de dados, 70% foram selecionadas para treino, 15% para teste e 15% para validação. Essa escolha foi feita por razões empíricas, e segue o padrão tipicamente adotado na literatura.

A Figura 1 contém exemplos das imagens contidas na base de dados. Enquanto as quatro primeiras colunas da esquerda são imagens de SNs reais, as quatro colunas remanescentes são artefatos (outros eventos que podem ser confundidos com SNs). Os números na parte inferior de cada imagem indicam a pontuação que cada detecção recebeu do algoritmo Autoscan, sendo 1,0 para classificação totalmente confiante de que a detecção é supernova, enquanto 0,0 indica o oposto.

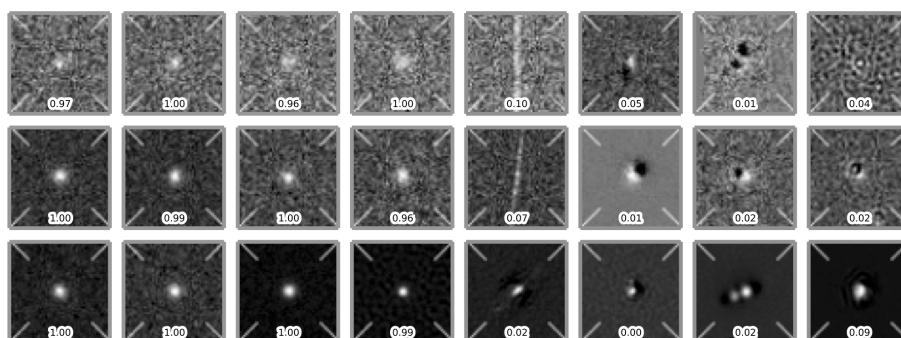


Figura 1. Diferentes imagens contidas na base de dados utilizada no Autoscan. Fonte: [Goldstein et al. 2015].

³<https://github.com/pippinlee/space2vec-ml-code>

⁴<https://github.com/guille-c/Deep-HiTS>

3.2. Arquitetura de Rede Neural Proposta

Neste trabalho, foi proposta uma arquitetura CNN para detectar SNs a partir de imagens de diferença. Toda a implementação foi feita na linguagem Python 3.6. O *hardware* usado contém um processador Intel(R) Core(TM) i7-9700KF CPU @ 3.60GHz e 16GB de memória RAM, com o sistema operacional Windows 10 64-bits. Na criação do modelo, utilizou-se o Tensorflow 2.2 para treinar, validar e testar a CNN. Foram realizados experimentos de modo a produzir o modelo com o melhor resultado. Devido às dimensões das imagens serem pequenas, optou-se por evitar o uso de muitas camadas de convolução, com o intuito de impedir a ocorrência de *overfitting*, que é o sobreajuste dos dados em relação ao modelo concebido, isto é, quando o modelo se ajusta muito bem aos dados do conjunto de treinamento mas se mostra ineficaz para analisar dados externos.

Ao todo, seis arquiteturas diferentes foram testadas, contendo uma ou duas camadas de convolução, além de 16, 32 ou 64 filtros. Também em razão do tamanho das imagens, foi escolhido um *kernel* de convolução 3×3 *pixels*. Usou-se a função de ativação ReLU para permitir rápida convergência de treinamento e teste, sem grandes diferenças de acurácia quando comparada a outras funções. Após a escolha da melhor arquitetura para o modelo, utilizou-se a validação cruzada para avaliar a capacidade de generalização de um modelo, dividindo a base de dados em 10 partes.

Para normalizar as entradas de dados e estabilizar o treinamento, também foi aplicado o *Batch Normalization* entre as camadas de convolução e *pooling*, o que permite acelerar o processo de treinamento da RNA. Optou-se por utilizar o *pooling* máximo (*Max Pooling*) de tamanho 2×2 *pixels*, pois ele realça os *pixels* mais brilhantes da imagem, que são os que se destacam e potencialmente ressaltam os padrões de uma supernova.

A seguir, utilizou-se uma camada totalmente conectada, com 128 neurônios e a função de ativação ReLU. O número de neurônios foi escolhido mantendo proporcionalidade com o tamanho da base de dados. Já a escolha de uma única camada densa decorre da necessidade de classificação sem que o modelo fique muito grande e possa causar o sobreajuste dos dados. Ao final, é utilizado o *dropout* de 20% após a camada densa. Essa técnica desativa aleatoriamente 20% dos neurônios utilizados no modelo, contribuindo para aumentar sua generalização. A camada de saída possui apenas um neurônio com a função de ativação *sigmoid*, por ser um problema de classificação binária. Para compilar o modelo, utilizou-se o otimizador *Adam* com uma taxa de aprendizado de 0,001, a função de perda *binary_crossentropy* e, como métrica de avaliação, *accuracy*.

Para garantir que o modelo seja treinado com o número de épocas ideal, utiliza-se de *callbacks* com parada antecipada (*early stopping*). Essa técnica possibilita interromper o treinamento caso não haja melhora dos resultados com o tempo. Como critério de parada para essa interrupção, foi verificado se não houve nenhuma diminuição do *val_loss* (valor de perda da função de custo do modelo durante a validação) em um período de três épocas.

Para avaliar os resultados, utilizam-se as métricas de Acurácia (proporção dos casos classificados corretamente), Sensibilidade (proporção de casos verdadeiros positivos), e Especificidade (proporção de casos verdadeiros negativos). Tais medidas são comumente utilizadas para avaliação de algoritmos de RNAs para problemas de classificação. A obtenção desses dados foi feita por meio de uma matriz de confusão, tipicamente voltada para compor a quantidade de falsos positivos (FP), de falsos negativos (FN), de ver-

dadeiros positivos (VP) e verdadeiros negativos (VN) dos resultados do modelo.

3.3. Otimização de Redes Neurais Artificiais

Para otimizar a arquitetura proposta, aplicou-se o *Tensorflow Model Optimization*, um conjunto de ferramentas do *Tensorflow* destinado a otimizar modelos de ML. O processo de otimização empregado consistiu em: (I) remoção dos pesos insignificantes, (II) remoção dos invólucros utilizados durante a poda e, por último, (III) aplicação de um algoritmo de compressão.

Na primeira etapa, aplicou-se o algoritmo de poda do *Tensorflow Model Optimization*, cuja finalidade é tornar nulos os parâmetros próximos a zero. A poda é aplicada ao modelo de RNA já treinado, o qual, após definidos os parâmetros, é novamente treinado. Quanto aos parâmetros da poda, optou-se por iniciar o modelo com 30% de esparsidade, ou seja, 30% dos parâmetros zerados e que são aumentados gradativamente, até finalizar o procedimento com 80%. A escolha por usar esse procedimento em todo o modelo em vez de em apenas algumas camadas foi feita com o objetivo de avaliar os efeitos em sua totalidade. Na etapa de treinamento do modelo podado, usou-se a mesma base de dados de treino e validação empregadas no treinamento inicial do modelo antes da poda. Também fez-se uso dos *callbacks* de parada antecipada.

Após o treinamento, o modelo retém vários parâmetros que são necessários apenas durante a poda. Esses invólucros foram removidos utilizando-se a função *strip_pruning* do *Tensorflow Model Optimization*, contribuindo para a diminuição de parâmetros desnecessários ao modelo. Em seguida, aplicou-se um algoritmo de compressão para a remoção dos pesos zerados, visto que esses são considerados redundantes. Finalmente, aplicou-se a função de compactação do pacote *zipfile* do Python, a fim de diminuir ainda mais o tamanho do modelo.

4. Resultados e Discussão

Nesta seção são apresentados os resultados obtidos em cada experimento realizado com as propostas de CNNs testadas para a Detecção de Supernovas. Por meio das métricas supracitadas, observou-se o desempenho de cada modelo antes e depois da poda e da compactação, bem como seu tamanho em bytes. Após apurar os resultados, é feita uma comparação dos mesmos aos da CNN desenvolvida pelo projeto Space2Vec, que utiliza a mesma base de dados.

Até o momento da escrita deste artigo, não foi possível comparar outra RNA treinada com a mesma base de dados usada aqui para que pudessem ser realizadas mais comparações diretas. A maioria dos trabalhos baseados em RNAs para detecção de SNs por imagens de diferença usam telescópios específicos e, portanto, bases de dados próprias, como é o caso de [Morii et al. 2016, Cabrera-Vives et al. 2017].

Na Tabela 1, são apresentados os resultados da etapa de teste dos algoritmos propostos e os do Space2Vec e Deep-HiTS. M1 representa o primeiro modelo, diferenciado dos demais por possuir uma única camada de convolução com 16 filtros. M2 é o modelo que possui duas camadas de convolução com 16 filtros cada. Já M3 possui uma camada de convolução com 32 filtros, enquanto M4 possui duas camadas de convolução com 32 filtros cada. M5 emprega uma camada de convolução com 64 filtros e M6, finalmente,

possui duas camadas de convolução com 64 filtros cada. As métricas referentes ao algoritmo do Space2Vec foram obtidas por meio da matriz de confusão disponibilizada em [Dessa 2019]. O tamanho do modelo em memória foi obtido por meio da representação do mesmo em um arquivo físico *h5*. Já a acurácia do Deep-HiTS foi disponibilizada em [Cabrera-Vives et al. 2017].

Modelo	Tamanho	ACC	SEN	ESP
M1	4,414 Megabytes	0,960	0,976	0,945
M1 Podado	1,399 Megabytes	0,950	0,957	0,944
M1 Podado e Compactado	1,340 Megabytes	0,950	0,957	0,944
M2	0,936 Megabytes	0,964	0,966	0,961
M2 Podado	0,300 Megabytes	0,953	0,969	0,939
M2 Podado e Compactado	0,297 Megabytes	0,953	0,969	0,939
M3	8,805 Megabytes	0,961	0,968	0,954
M3 Podado	2,804 Megabytes	0,959	0,964	0,954
M3 Podado e Compactado	2,803 Megabytes	0,959	0,964	0,954
M4	1,891 Megabytes	0,963	0,971	0,956
M4 Podado	0,606 Megabytes	0,962	0,967	0,957
M4 Podado e Compactado	0,592 Megabytes	0,962	0,967	0,957
M5	17,645 Megabytes	0,958	0,979	0,939
M5 Podado	5,605 Megabytes	0,945	0,947	0,944
M5 Podado e Compactado	5,285 Megabytes	0,945	0,947	0,944
M6	3,848 Megabytes	0,967	0,978	0,955
M6 Podado	1,223 Megabytes	0,964	0,969	0,959
M6 Podado e Compactado	1,190 Megabytes	0,964	0,969	0,959
CNN space2vec	6,309 Megabytes	0,924	0,990	0,857
Deep-HiTS	-	0,994	0,996	-

Tabela 1. Resultados da etapa de teste dos 6 modelos propostos, comparados ao Space2Vec e Deep-HiTS.

O uso da parada antecipada foi importante para mitigar o sobreajuste dos dados decorrentes das épocas de treinamento. Realizou-se um experimento para observar o comportamento da rede sem a parada antecipada, treinando a arquitetura M6 com 30 épocas. Pode-se perceber uma possível ocorrência do sobreajuste através do gráfico da função de perda do treino e validação (Figura 2), que começou a divergir a partir da quinta época.

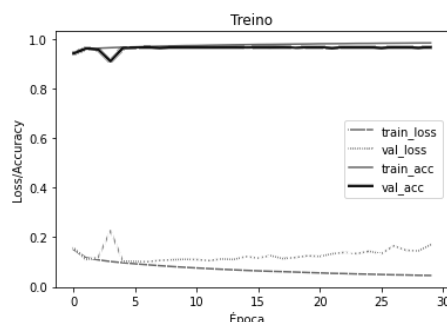


Figura 2. Gráfico da convergência de M6 durante o treinamento sem o uso da parada antecipada.

Dentre as opções, a arquitetura M6 obteve os melhores resultados no teste, alcançando uma acurácia de 0,967. Na etapa de treinamento, os resultados obtidos na última época foram: acurácia de 0,974 e perda de 0,074 para o treino, enquanto a

validação obteve 0,965 para a acurácia e 0,104 para a perda. Neste modelo, após aplicar a poda, houve apenas 0,3% de perda de acurácia, enquanto a diminuição do tamanho do modelo em bytes foi de 69%. A aplicação do algoritmo da poda manteve resultados satisfatórios para todos os modelos testados, tornando-os até 70% menores. A perda de acurácia decorrente da poda em toda a RNA foi mínima, variando de 0,15% de perda no melhor caso à 1,35% no pior. Os gráficos apresentados na Figura 3 representam o comportamento das métricas de acurácia e perda durante o treinamento e validação do Modelo 6 com e sem poda.

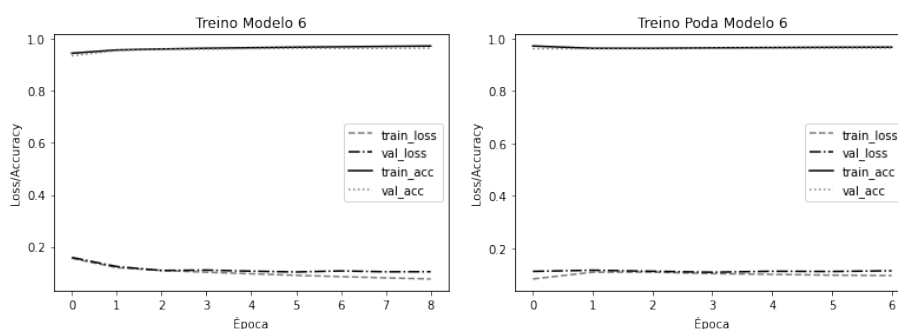


Figura 3. Gráficos de treinamento e poda do Modelo 6. A imagem da esquerda mostra o comportamento da arquitetura em sua forma original e, a direita, o mostra após a poda.

Utilizando a arquitetura M6, também treinou-se o modelo e a poda com a Validação Cruzada, dividindo a base de dados em 10 partes, alternando-as entre Treino e Teste. A escolha do número de épocas para o treino baseou-se nos testes realizados no experimento anterior, com conjunto de Validação e Parada Antecipada. O modelo foi treinado com sete épocas, enquanto a poda foi treinada com seis épocas. A acurácia média obtida pelo modelo foi de 0,963, com o desvio padrão de 0,003. Já a poda obteve uma acurácia média de 0,972 com o desvio padrão de 0,002.

Comparando M6 com o algoritmo do Space2Vec, observou-se que o modelo proposto obteve uma acurácia maior (0,964 contra 0,924) e o seu tamanho em disco é 80% menor, uma diferença significativa. Comparou-se também os resultados de M6 com os obtidos pelo algoritmo Deep-HiTS [Cabrera-Vives et al. 2017], que apesar de não usar a mesma base de dados, soluciona o mesmo problema de detectar SNs. Enquanto a acurácia do nosso modelo é de 0,964, a apresentada pelo Deep-HiTS foi de 0,994. Porém, nosso modelo, por possuir menos camadas, apresenta uma complexidade muito menor, facilitando os cálculos e tornando sua execução mais rápida.

Conclui-se que o modelo descrito neste trabalho é bastante promissor para o problema de Detecção de Supernovas, equiparando-se com outros trabalhos existentes, e que a aplicação da poda possibilita um aprimoramento. Isto favorece a execução do mesmo em dispositivos com poucos recursos de *hardware* sem que haja piora significativa nos resultados produzidos.

Para trabalhos futuros, almeja-se comparar outras propostas existentes com a nossa arquitetura, considerando a mesma base de dados. Além disso, é desejável verificar o comportamento da rede quando a poda é aplicada em camadas específicas, em vez de toda a rede. Dado que existem camadas que impactam mais severamente no resultado

do que outras, escolher camadas específicas para a poda pode proporcionar um melhor desempenho do modelo.

Referências

- Burrows, A. and Goshy, J. (1993). A theory of supernova explosions. *The Astrophysical Journal*, 416:L75.
- Burrows, A. and Vartanyan, D. (2021). Core-collapse supernova explosion theory. *Nature*, 589(7840):29–39.
- Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., and Maureira, J.-C. (2017). Deep-hits: Rotation invariant convolutional neural network for transient detection. *The Astrophysical Journal*, 836(1):97.
- Dessa (2019). space2vec: Identifying supernovas with deep learning. Disponível em: <https://medium.com/dessa-news/space-2-vec-fd900f5566>. Acesso em: 18 mar 2021.
- Flaugher, B. (2005). The dark energy survey. *International Journal of Modern Physics A*, 20(14):3121–3123.
- Förster, F., Maureira, J. C., San Martín, J., Hamuy, M., Martínez, J., Huijse, P., Cabrera, G., Galbany, L., De Jaeger, T., González-Gaitán, S., et al. (2016). The high cadence transient survey (hits). i. survey design and supernova shock breakout constraints. *The Astrophysical Journal*, 832(2):155.
- Goldstein, D. A., D’Andrea, C. B., Fischer, J. A., Foley, R. J., Gupta, R. R., Kessler, R., Kim, A. G., Nichol, R. C., Nugent, P. E., Papadopoulos, A., and et al. (2015). Automated transient identification in the dark energy survey. *The Astronomical Journal*, 150(3):82.
- Ishida, E. E. (2019). Machine learning and the future of supernova cosmology. *Nature Astronomy*, 3(8):680–682.
- LeCun, Y., Denker, J. S., Solla, S. A., Howard, R. E., and Jackel, L. D. (1989). Optimal brain damage. In *NIPs*, volume 2, pages 598–605.
- Morii, M., Ikeda, S., Tominaga, N., Tanaka, M., Morokuma, T., Ishiguro, K., Yamato, J., Ueda, N., Suzuki, N., Yasuda, N., and Yoshida, N. (2016). Machine-learning selection of optical transients in the Subaru/Hyper Suprime-Cam survey. *Publications of the Astronomical Society of Japan*, 68(6). 104.
- Perlmutter, S., Aldering, G., Goldhaber, G., Knop, R., Nugent, P., Castro, P. G., Deustua, S., Fabbro, S., Goobar, A., Groom, D. E., et al. (1999). Measurements of ω and λ from 42 high-redshift supernovae. *The Astrophysical Journal*, 517(2):565.
- Riess, A. G., Filippenko, A. V., Challis, P., Clocchiatti, A., Diercks, A., Garnavich, P. M., Gilliland, R. L., Hogan, C. J., Jha, S., Kirshner, R. P., et al. (1998). Observational evidence from supernovae for an accelerating universe and a cosmological constant. *The Astronomical Journal*, 116(3):1009.
- Sobrinho, J. (2013). Estrelas: origem, evolução e morte. *Formação Contínua de docentes: Introdução à Astronomia (texto de apoio ao módulo 3)*, 17pp, Universidade da Madeira.